UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS

FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

E.A.P. DE INGENIERÍA DE SISTEMAS

Un modelo de Razonamiento Basado en Casos para la Captación de Requisitos en el desarrollo de proyectos de software

TESIS

Para optar el Grado de Magíster en Ingeniería de Sistemas con mención de Ingeniería de Software

AUTOR

Lenis Rossi Wong Portillo

ASESOR

Dr. David Mauricio Sánchez

Lima – Perú

2012

Este trabajo está dedicado a toda mi familia, en especial a mi hija Siulei. A mi esposo por su incondicional apoyo, compresión y amor. A mis padres y hermanos a quienes quiero mucho.

AGRADECIMIENTOS

Al profesor Mg. Cayo León Fernández y a la profesora Mg. Virginia Vera Pomalaza, por darme la oportunidad de realizar y concluir mis estudios de maestría.

Al profesor Dr. David Mauricio Sánchez, por su apoyo, orientación, dedicación y esfuerzo para que este trabajo de investigación cumpla con los objetivos trazados.

Al profesor Dr. Erick Papa Quiroz, por su orientación, consejos y revisiones del presente trabajo.

A todos mis profesores de la maestría, por la calidad de los conocimientos transmitidos y el legado recibido, por sus observaciones y porque en todo momento me incentivaron para que culmine este trabajo.

A todos mis colegas y amigos que indirectamente me ayudaron para culminar este trabajo y que muchas veces constituyen un invalorable apoyo.

Y, por encima de todo, doy gracias a Dios, por darme la oportunidad de culminar este trabajo.

Un modelo de Razonamiento Basado en Casos para la Captación de Requisitos en el desarrollo de proyectos de software

RESUMEN

Los estudios muestran que una de las causas de los atrasos de los proyectos de software se encuentra en la Captación de Requisitos. Se sabe que el costo para reparar un error en los requisitos es 5 a 10 veces menos que en la codificación y 200 veces menos que en el mantenimiento.

Existen muchos esfuerzos para mitigar este problema, como por ejemplo: herramientas y técnicas que facilitan el trabajo en la administración y gestión de requisitos, tales como REM, Entrevistas, Encuestas, Casos de Usos, Win Win, etc. Sin embargo, según los últimos reportes aún se mantienen los problemas.

Es por ello que en el presente trabajo proponemos *Un Modelo de Razonamiento Basado en Casos para la Captación de Requisitos en el desarrollo de proyectos de software.* Con esta técnica de inteligencia artificial se aprovecha los requisitos funcionales de proyectos de software desarrollados anteriormente, para luego resolver o identificar requisitos funcionales de un nuevo proyecto.

Se realizan dos casos de estudio donde se compara el *Modelo CBR* propuesto con otras dos técnicas: C*asos de Usos y Win Win*, se aplica métricas de calidad a los requisitos obtenidos con el modelo propuesto, Casos de Uso y Win Win, finalmente, se ve el éxito de los resultados.

Palabras clave: Razonamiento basado en casos, Requisito, Captación de Requisitos.

A model of case-based reasoning to capture requirements in the development of software projects

ABSTRACT

Studies show that one of the causes of delay in software projects is in the Collection Requirements. It is also known that the cost to repair an error in the requirements is 5 to 10 times less than in the encoding and 200 times less than in maintenance.

Thus there are many efforts to mitigate this problem. Such as: tools and techniques that make working in the administration and management requirements, such as REM, Interviews, Surveys, Case Uses, Win Win, etc. However according to the latest reports are the problems.

That is why in this paper we propose a model of Case-Based Reasoning for Collecting Requirements in the software development project. With this artificial intelligence technique, is used the functional requirements of software projects previously developed to solve or identify functional requirements of a new project.

There are two case studies, which compares the proposed CBR model with two other techniques: Use Cases and Win Win. Quality Metrics Applicable requirements obtained with the proposed model, Use Cases and Win Win, finally see the successful outcome.

Keywords: Case-based reasoning, Requirement, Requirements Elicitation.

ÍNDICE GENERAL

Lista de figuras

Lista de tablas

CAPÍTULO I

INTRODUCCIÓN

1.1 Antecedentes

1.1.1 Del Problema

Una de las causas de los fracasos de los proyectos de software se encuentra en la Captación de Requisitos. Según el reporte del caos del año 1995 [StandishGroup 1995], sólo el 16% de los proyectos son acabados con éxito, el 53% son terminados con problemas y el 31% son cancelados. En este reporte se puede observar que los *Requerimientos incompletos* representan el 13.1 % de la causa de los fracasos (ver Figura 1.1).

Además, se sabe que el costo para reparar un error en los requisitos es 5 a 10 veces menos que en la codificación y 200 veces menos que en el mantenimiento [Boehm+ 2010].

Para mitigar este problema existen muchas herramientas y técnicas que facilitan el trabajo en la captación, validación y la administración de los requisitos, tales como: Brainstorming [Raghavan+ 1994], *Entrevistas* [Piattini+ 1996], *Casos de Usos* [Booch+ 1999], *Encuestas* [Andriano 2006], *Prototipación* [Andriano 2006], *Win Win* [Chao+ 2009] y *JAD* [Dube+ 2010], entre otras. Sin embargo, según el reporte del caos del año 2009, siguen los problemas [StandishGroup 2009].

1.1.2 De la Técnica

El Razonamiento basado en casos (CBR) tiene sus raíces en el trabajo de Roger Schank y sus estudiantes en la Universidad de Yale a principio de la década de los 80. El modelo de memoria dinámica de Schank fue la base de los tempranos sistemas CBR. Otra escuela de Razonamiento basado en casos también emergió en la década del 80, investigando los asuntos tales como Razonamiento basado en casos en el razonamiento legal, el razonamiento basado en la memoria (una manera de razonar con ejemplos en las máquinas masivamente paralelas), y combinaciones de Razonamiento basado en casos con otros métodos de razonamiento [Althoff 2001].

En los años 90, el interés en Razonamiento basado en casos creció en la comunidad internacional, según se pudo evidenciar en el establecimiento de una conferencia internacional sobre el Razonamiento basado en casos en 1995, en conferencias europeas, ya sea alemanas, británicas, italianas y otros talleres de Razonamiento basado en casos. La tecnología Razonamiento basado en casos produjo numerosos casos de éxito [Althoff 2001].

Como han ido pasando los años, CBR ha sido utilizada para crear numerosas aplicaciones en una amplia gama de dominios, tales como: en el Análisis Financiero, Evaluación de Riesgo, Planificación, Mantenimiento Técnico, Proceso de Control, Control de Calidad, Diagnóstico Médico, Planificación, Diseño, Comercio Electrónico, Gestión del Conocimiento, Ingeniería de Software, etc. [Althoff 2001].

1.2 Definición del Problema

Una de las causas de los fracasos de los proyectos de software se encuentra en la Captación de Requisitos. Esto conlleva a que los proyectos no sean terminados con éxito, ocasionando demoras o atrasos en la entrega de los proyectos y en varias ocasiones estos proyectos son cancelados.

1.3 Justificación

1.3.1 Justificación Práctica

Los estudios muestran que una de las causas de los fracasos de los proyectos de software se encuentra en los Requisitos. Según el reporte del caos publicado el año 1995 [StandishGroup 1995], no todos los proyectos de software son terminados con éxito. Sólo el 16% de los proyectos son terminados con éxito, el 53% son terminados con problemas y el 31% son cancelados, según datos del año 1994 (ver Tabla 1.2).

En este reporte se puede observar que existen muchos factores que causan los fracasos de estos proyectos, uno de ellos es *Requisitos incompletos*. Según la Figura 1.1, los *Requerimientos incompletos* representan el 13.1 % de la causa de los fracasos.

Figura 1. 1. Reporte del Caos, datos estadísticos del año 1994 [StandishGroup 1995].

Para mitigar este problema han surgido muchas herramientas y técnicas que facilitan el trabajo en la administración de requisitos, tales como: Brainstorming [Raghavan+ 1994], Entrevistas [Piattini+ 1996], Casos de Usos [Booch+ 1999], Encuestas [Andriano 2006], Prototipación [Andriano 2006], Win Win [Chao+ 2009] y JAD [Dube+ 2010], entre otras.

Sin embargo, a pesar de ello siguen los problemas. En el Reporte del Caos del año 2009 [StandishGroup 2009] los resultados no son alentadores. El 32 % de los proyectos son terminados con éxito, el 44% son terminados con problemas y el 24% son cancelados. Así pues, se puede observar un aumento en la cantidad de proyectos cancelados en comparación con los datos del año 2004.

Tabla 1. 2. Reporte del Caos, datos estadísticos por años [StandishGroup 2009].

Proyectos	**1994**	**1998**	**2000**	**2002**	**2004**	**2009**
Exitosos	16%	26%	28%	34%	29%	32%
Cancelados	31%	28%	23%	15%	18%	24%
Con problemas	55%	46%	49%	51%	53%	44%

1.3.2 Justificación Teórica

El Razonamiento Basado en Casos (CBR) es una técnica de Inteligencia Artificial que permite aprovechar la experiencia acumulada en la solución de nuevos problemas. Con esta técnica se almacenan casos con la solución que se ha dado anteriormente y cuando se presenta un nuevo problema esta información o experiencia acumulada es empleada para resolverlo [Althoff 2001].

Los CBR se han aplicado con éxito en el campo académico, en la industria, en la medicina y en la ingeniería. Por ejemplo, en el desarrollo de herramientas que ayuden en la planificación: CHEF [Hammond 1986], en los diagnósticos: PROTOS [Bareiss 1998], en el Diseño: KRITIK [Goel 1989], en la industria: FormTool [Juárez+ 2005], etc.

También se han aplicado CBR con éxito en la Ingeniería de Software, tales como: en la *Optimización de la calidad del software* [Brady+ 2010], en *el Diseño de Controles de Sistemas de Información basados en Internet* [Lee+ 2009], en la *Estimación de los esfuerzos en el desarrollo de proyectos de software* [Mendes+ 2002], etc.

1.4 Objetivos

1.4.1 Objetivo General

El objetivo de esta tesis es diseñar un modelo para la Captación de Requisito en el desarrollo de un proyecto de software aplicando Razonamiento Basado en Casos (CBR).

1.4.2 Objetivos Específicos

Asimismo, basándonos del objetivo general, se plantea los siguientes objetivos específicos:

- OE1: Analizar técnicas existentes para la Captación de Requisitos.
- OE2: Estudiar modelos de Razonamiento Basado en Casos existentes.
- OE3: Escoger el modelo de Razonamiento Basado en Casos más adecuado para nuestra propuesta.
- OE4: Conocer herramientas existentes en el mercado para el desarrollo de sistemas de Razonamiento Basado en Casos.
- OE5: Analizar varios trabajos donde se hayan aplicado CBR en la ingeniería de Software.
- OE6: Desarrollar una herramienta para poder implementar el modelo propuesto basándonos del modelo CBR más adecuado para nuestro modelo.
- OE7: Desarrollar dos casos de estudio para verificar que el Modelo propuesto tiene éxito.
- OE8: Aplicar métricas de calidad a los resultados obtenidos.

1.5 Alcances

La presente tesis tendrá los siguientes alcances:

- El modelo que se propone ayudará a la Captación de Requisitos en el desarrollo de proyectos de software.

- Se realizará el estado del arte de modelos, técnicas y herramientas relacionados con CBR.

- Se realizará el estado del arte de técnicas, métodos y herramientas para la Captación de Requisitos.

- Se realizará también el estado del arte de las últimas investigaciones de CBR en la ingeniería de software.

- Se desarrollará una herramienta que nos permitirá implementar el modelo propuesto.

- Se realizará dos casos de estudios donde se probará el modelo propuesto frente a otras técnicas de captación de requisitos tradicional.

- Se aplicará cuatro métricas de calidad a los resultados obtenidos.

1.6 Estructura de la tesis

La presente tesis está organizada en 7 capítulos que a continuación se explican brevemente:

En el Capítulo 2 se describe el marco teórico, donde se explican los principales conceptos asociados con el modelo propuesto, los cuales son abordados a lo largo de la presente investigación. Son parte de este capítulo: las definiciones de la Ingeniería de Requisitos, Requisitos, características, definiciones del Razonamiento Basado en Casos, tipos, Ciclo de vida CBR, Sistemas CBR, Herramientas para el desarrollo de SRBC, Marcos conceptuales y glosario, etc.

En el Capítulo 3 se describe el estado del arte, donde se detallan las técnicas existentes para la Captación de Requisitos y se analiza cada una de estas técnicas. También se describen las herramientas para la Captación de Requisitos y luego se presentan las últimas investigaciones sobre el uso de CBR en la Ingeniería de Software, tales como: en la Estimación de costos de proyectos de software utilizando la herramienta CBR-Works [Mendes+ 2002], en el Diseño de Controles de Sistemas de Información basados en Internet [Lee+ 2009] y en la Optimización de la calidad

del software [Brady+ 2010]. De este modo, se brinda el marco sobre el cual se formula la presente investigación.

En el Capítulo 4 se describe el aporte teórico, donde se presenta el modelo CBR propuesto; se describe la justificación del uso de Razonamiento Basado en Casos, se presenta el mapa del modelo y, por último, se expone la descripción general y específica del modelo CBR propuesto. Dicho modelo debidamente justificado, quedará expedito para su respectiva validación.

En el Capítulo 5 se presenta una herramienta que permitirá la implementación del modelo CBR propuesto; donde se describe cómo se ha construido esta herramienta y se explica los pasos a seguir para el uso de la herramienta.

En el Capítulo 6 se describe el aporte práctico, donde se presenta dos casos de estudios. El primer caso de estudio trata sobre el Sistema Integrado de Recursos Humanos (SIRH), que se desarrolló para la Caja de Pensiones Militar Policial. El segundo caso de estudio trata sobre el Sistema de Punto de Venta (SPTA), que se desarrolló para la cadena de farmacias Boticas FASA. Primero se aplicará a cada caso de estudio dos métodos de captación de requisitos y el modelo CBR propuesto. Luego se aplicará métricas de calidad a los requisitos obtenidos con cada técnica y el Modelo CBR propuesto, y por último se compararán los resultados.

Finalmente, en el Capítulo 7, se expone la conclusión general y las conclusiones específicas de acuerdo a cada objetivo específico, y luego se presentan las recomendaciones y trabajos futuros.

CAPÍTULO 2

MARCO TEÓRICO

2.1 Ingeniería de Requisitos

2.1.1 Concepto

Existen muchas definiciones de Ingeniería de requisitos o requerimientos, algunas de ellas son:

- La Ingeniería de requisitos trata de los principios, métodos, técnicas y herramientas que permiten descubrir, documentar y administrar los requisitos del software, de forma sistemática y repetible. Es la rama de la Ingeniería de Software relacionada con los objetivos del mundo real, las funciones y restricciones de los sistemas de software [Andriano 2006].

- La Ingeniería de requisitos es la disciplina para desarrollar una especificación completa, consistente y no ambigua, la cual servirá como base para acuerdos comunes entre todas las partes involucradas y en donde se describen las funciones que realizará el sistema [Boehm + 1979].

- "Es el proceso mediante el cual se intercambian diferentes puntos de vista para recopilar y modelar lo que el sistema va a realizar. Este proceso utiliza una combinación de métodos, herramientas y actores, cuyo producto es un modelo del cual se genera un documento de requisitos" [Leite 1987].

- "Ingeniería de requisitos es un enfoque sistémico para recolectar, organizar y documentar los requisitos del sistema; es también el proceso que establece y

mantiene acuerdos sobre los cambios de requisitos, entre los clientes y el equipo del proyecto" [Oberg+ 2000].

2.1.2 Importancia

Los principales beneficios que se obtienen de la Ingeniería de Requisitos son [Andriano 2006]:

- Permite gestionar las necesidades del proyecto en forma estructurada: cada actividad de la Ingeniería de Requisitos consiste de una serie de pasos organizados y bien definidos.

- Mejora la capacidad de predecir cronogramas de proyectos, así como sus resultados: la Ingeniería de Requisitos proporciona un punto de partida para controles subsecuentes y actividades de mantenimiento, tales como: estimación de costos, tiempo y recursos necesarios.

- Disminuye los costos y retrasos del proyecto: es sabido que reparar errores por un mal desarrollo no descubierto a tiempo es sumamente caro, especialmente aquellas decisiones tomadas durante la Ingeniería de Requisitos, ya que es una de las etapas de mayor importancia en el ciclo de desarrollo de software y de las primeras en llevarse a cabo.

- Mejora la calidad del software: la calidad en el software tiene que ver con cumplir un conjunto de requisitos (funcionalidad, facilidad de uso, confiabilidad, desempeño, etc.).

2.1.3 Actividades de la Ingeniería de Requisitos

La Ingeniería de Requisitos está compuesta por actividades que involucran el desarrollo y la administración de los requisitos, tal como se muestra en la Figura 2.1 [Andriano 2006].

El desarrollo de requisitos involucra cuatro actividades básicas, tales como: captación, análisis, especificación y verificación. Por su parte, la administración de

requisitos está compuesta por las actividades: control de cambios, control de versiones, monitoreo de requisitos y monitoreo del estado de los requisitos (ver Figura 2.1) [Andriano 2006]:

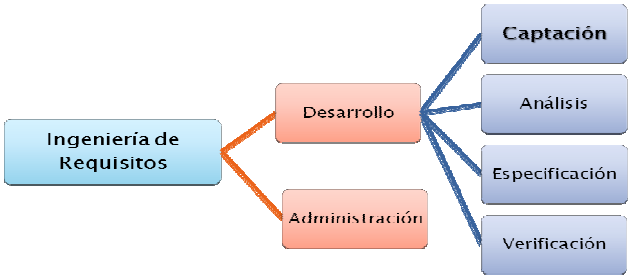


Figura 2. 1. Ingeniería de Requisitos, Tareas [Andriano 2006].

- **Captura**

Esta fase representa el comienzo de cada ciclo. La captura o extracción es el nombre comúnmente dado a las actividades involucradas en el descubrimiento de los requisitos del sistema. Aquí los analistas de requisitos deben trabajar junto al cliente para descubrir el problema que el sistema debe resolver, los diferentes servicios que el sistema debe prestar y las restricciones que se pueden presentar. Es importante que la captura sea efectiva, ya que la aceptación del sistema dependerá de cuan bien éste satisfaga las necesidades del cliente.

- **Análisis**

Sobre la base de la Captura realizada previamente comienza esta fase, la cual se enfoca en descubrir problemas con los requisitos del sistema identificados hasta el momento. Usualmente se hace un análisis luego de haber producido un bosquejo inicial del documento de requisitos; en esta etapa se leen los requisitos, se conceptúan, se investigan, se intercambian ideas con el resto del equipo, se resaltan los problemas, se buscan alternativas y soluciones, y luego se van fijando reuniones con el cliente para discutir los requisitos.

- **Especificación**

En esta fase se documentan los requisitos acordados con el cliente, en un nivel apropiado de detalle. En la práctica, esta etapa se va realizando conjuntamente con el análisis, se puede decir que la especificación es el "pasar en limpio" el análisis realizado previamente aplicando técnicas y/o estándares de documentación, como la notación UML (Lenguaje de Modelado Unificado), que es un estándar para el modelado orientado a objetos, por lo que los casos de uso y la obtención de requisitos basada en casos de uso se utiliza cada vez más para la obtención de requisitos.

- **Validación**

La validación es la etapa final de la Ingeniería de Requisitos. Su objetivo es verificar todos los requisitos que aparecen en el documento especificado para asegurarse que representan una descripción, por lo menos, aceptable del sistema que se debe implementar. Esto implica verificar que los requisitos sean consistentes y que estén completos.

Es necesario recalcar que no existe un proceso único que sea válido de aplicar en todas las organizaciones. Cada organización debe desarrollar su propio proceso de acuerdo al tipo de producto que se esté desarrollando, a la cultura organizacional, y al nivel de experiencia y habilidad de las personas involucradas en la Ingeniería de Requisitos.

Hay muchas maneras de organizar el proceso de ingeniería de requisitos y en otras ocasiones se tiene la oportunidad de recurrir a consultores, ya que ellos tienen una perspectiva más objetiva que las personas involucradas en el proceso.

2.1.4 Requisitos

Entre las principales definiciones de Requisito o Requerimientos de un sistema de software tenemos [IEEE 1990]:

- Una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo.

- Una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal.

- Una representación documentada de una condición o capacidad.

2.1.5 Tipos

Los requisitos o requerimientos de software se dividen en dos categorías [Grillo+ 2009]:

- **Funcionales.** Los requisitos funcionales son los que definen las funciones que el sistema será capaz de realizar, además, describen las transformaciones que el sistema realiza sobre las entradas para producir salidas. Es importante que se describa el ¿Qué? y no el ¿Cómo? se deben hacer esas transformaciones.

- **No funcionales.** Son las características que de una u otra forma limitan el sistema, algunos de ellos son: el rendimiento (en tiempo y espacio), interfaces de usuario, fiabilidad (robustez del sistema y disponibilidad de equipo), mantenimiento, seguridad, portabilidad y estándares.

2.1.6 Características

Las principales características de los requisitos son las siguientes [Grillo+ 2009]:

- **Necesario.** Un requerimiento es necesario si su omisión provoca una deficiencia en el sistema a construir y no pueden ser reemplazados por otras capacidades del producto o del proceso.

- **Conciso.** Un requerimiento es conciso si es fácil de leer y entender. Su redacción es simple y clara.

- **Completo.** Un requerimiento está completo si no necesita ampliar detalles en su redacción y si proporciona la información suficiente para su comprensión.

- **No ambiguo.** Un requerimiento no es ambiguo cuando tiene una sola interpretación.

- **Verificable**. Un requerimiento es verificable cuando puede ser cuantificado a través de los siguientes métodos de verificación: inspección, análisis, demostración o pruebas.

2.1.7 Dificultad en la Definición de los Requisitos

Algunas de las dificultades que presentan los requisitos se pueden observar a continuación [Grillo+ 2009]:

- Los requisitos vienen de muchas fuentes.

- Hay gran variedad en los tipos de requisitos.

- No son iguales. Algunos son más difíciles, más riesgosos, importantes o toman más tiempo que otros.

- Los requisitos están relacionados unos con otros y a su vez se relacionan con otras partes del proceso.

- Un requisito puede cambiar a lo largo de su ciclo de vida.

Debido a la ausencia de requisitos se pueden presentar problemas durante el desarrollo de un sistema, estos problemas son:

- No se entienden los problemas del usuario.

- No se documentan las necesidades del usuario.

- Falta de acuerdos entre el desarrollador y el usuario.

- No se controlan los cambios del requerimiento.

- Falta definición y delimitación de responsabilidades de las partes involucradas.

2.1.8 Consecuencias en la mala identificación de Requisitos

Cuando los requisitos o requerimientos identificados en el análisis del desarrollo de un proyecto de software tienen errores ocurre lo siguiente [Grillo+ 2009]:

- El sistema se entregará tarde y el costo real será mayor al planificado.
- Los clientes y usuarios estarán disconformes con el sistema.
- El sistema será considerado poco confiable.
- Si el sistema se mantiene en uso, los costos de mantenimiento serán muy elevados.

2.2 Razonamiento Basado en Casos (CBR)

2.2.1 Concepto

El Razonamiento basado en Casos (CBR) es definido como "una técnica de la inteligencia artificial que intenta llegar a la solución de nuevos problemas de forma similar como lo hacen los seres humanos utilizando la experiencia acumulada hasta el momento en acontecimientos similares" [Rossillea+ 1993].

Un nuevo problema se compara con los casos almacenados previamente en la base de casos (Memoria de Casos) y se recuperan uno o varios casos. Posteriormente, se utiliza y evalúa una solución sugerida por los casos que han sido seleccionados con anterioridad para tratar de aplicarlos al problema actual [Juárez+ 2005].

Básicamente, el CBR es un modelo de razonamiento que nos permite resolver problemas, entender situaciones y aprender, todo ello integrado con los mecanismos asociados a la memoria [Althoff 2001].

Este modelo nos ofrece un nuevo paradigma para la construcción de sistemas inteligentes que se basa en la utilización de la experiencia previa. De esta manera, podemos construir razonadores que resuelven nuevos problemas mediante la

adaptación de soluciones que han sido utilizadas para resolver problemas pasados [Juárez+ 2005].

2.2.2 Caso

Un *caso* es una situación concreta que describe un problema. Se corresponde con una situación previamente experimentada, capturada y aprendida, de tal forma que puede ser reutilizada en la resolución de problemas futuros [Aamodt+ 1994].

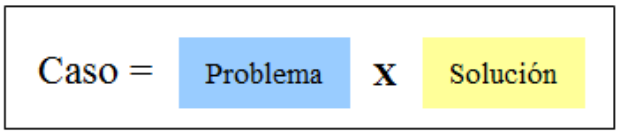


Figura 2.2. Descripción de un caso [Juárez+ 2005].

Un caso está compuesto por la descripción del problema y la solución de dicho problema (ver Figura 2.2). A su vez, un problema está compuesto por un conjunto de atributos finitos que determinan los valores de las diferentes características que definen el problema.

Por lo tanto, el problema descrito de un caso se puede considerar como un vector que contienen los valores que toman dichos atributos. Estos atributos pueden ser cualitativos o cuantitativos dependiendo de la naturaleza del problema, donde la solución del problema puede ser uno o más atributos del problema (ver Figura 2.3) [Juárez+ 2005].

$$\text{Problema} = \{ at1, at2, at3, \ldots at_n \}$$

Figura 2.3. Descripción de un problema [Juárez+ 2005].

2.2.3 Sistema de Razonamiento Basado en Casos (SRBC)

Un Sistema de Razonamiento Basado en Casos (SRBC) almacena un conjunto de problemas ya resueltos y recibe como entrada un nuevo problema. El sistema

intentará resolver el problema de entrada buscando, de entre los problemas resueltos, el más semejante y adaptando su solución al problema de entrada [Juárez+ 2005].

A diferencia de otras aproximaciones de la IA, el CBR no depende únicamente del conocimiento general del dominio del problema, sino que utiliza el conocimiento específico de casos pasados para reutilizarlo como solución del nuevo problema. De esta manera, se evita el cuello de botella que supone la adquisición de conocimiento para el desarrollo de CBR [Juárez+ 2005].

2.2.4 Ventajas de un SRBC

¿Qué ventajas ofrecen los SRBC respecto a los sistemas tradicionales de resolución de problemas? Existen cinco ventajas frente a los sistemas expertos tradicionales [Juárez+ 2005]:

- ***Adquisición de Conocimiento***

Es bien sabido el cuello de botella que supone el proceso de adquisición de conocimiento en los SBC tradicionales. El proceso de extracción de reglas a partir de un experto requiere un enorme esfuerzo que no garantiza la validez de dichas reglas.

Los SRBC no requieren este proceso, pues utilizan experiencias previas para resolver casos. En los dominios donde la resolución de problemas se basa en experiencias previas (por ejemplo, ciertos campos de la medicina y la jurisprudencia), el coste de la adquisición de conocimiento para un SRBC es mínima.

Sin embargo, esto no es aplicable a todos los dominios, ya que los casos pueden no estar disponibles, ser difíciles de usar o estar incompletos. En estas situaciones, el desarrollo de un SRBC depende de lo costoso que sea la ingeniería de los casos: determinar la información que deben incluir, elegir la representación adecuada y extraer dicha información a partir de los datos disponibles.

- ***Mantenimiento del Conocimiento***

En los SBC tradicionales el mantenimiento de la base de conocimiento (BC) es una de las tareas más costosas del proceso de desarrollo (aparte del ya mencionado cuello de botella de la adquisición de conocimiento).

Normalmente, el contenido inicial de la BC se ve sujeto a la adición del nuevo conocimiento, lo cual puede implicar la redefinición del conocimiento existente e incluso se puede dar el caso de que éste llegue a estar obsoleto.

Los SRBC permiten que los usuarios añadan nuevos casos a la librería de casos sin intervención del experto (aunque tiene que existir algún mecanismo que valide los nuevos casos). Además, gracias a que los SRBC realizan un aprendizaje incremental, inicialmente pueden empezar a trabajar con un conjunto limitado de casos e ir incorporando nuevos casos a medida que sea necesario.

- ***Eficiencia en la Resolución de Problemas***

La reutilización de casos previos permite que se puedan resolver problemas similares sin tener que rehacer el proceso de razonamiento. Además, debido a que los SRBC pueden almacenar casos fallidos, estos pueden servir como avisos para futuros problemas a evitar.

- ***Calidad de la Solución***

Existen situaciones en las que las BC de los SBC tradicionales pueden tener reglas imperfectas, especialmente si no se ha entendido correctamente el dominio de aplicación. En estas circunstancias, las soluciones proporcionadas por los casos son más precisas, ya que reflejan qué es lo que realmente ha sucedido en un contexto determinado.

- ***Aceptación del Usuario***

Para que un sistema inteligente sea aceptado por los usuarios, éstos deben confiar en las soluciones que aportan. Para ello resulta extremadamente importante que los usuarios confíen en el proceso que lleva a dichas soluciones.

2.2.5 Elementos para la construcción de un CBR

Según [Shiu+ 2004], hay tres temas importantes que se tiene que tomar en cuenta en la construcción de un sistema CBR: la representación de los casos, la indexación de los casos y el modelo de memoria usado para la librería de casos.

2.2.6 Representación de Casos

Los casos en LC pueden ser representados en diferentes tipos de conocimientos y se pueden almacenar en diferentes formatos. En cada tipo de sistema CBR, un caso puede representar algo diferente. Por ejemplo, los casos podrían representar a las personas, objetos, situaciones, diagnósticos, diseños, planos, o resoluciones, entre muchas otras representaciones [Shiu+ 2004].

Todo va depender del tipo de sistema CBR. Por ejemplo, en esta investigación el caso va representar un *requisito* funcional de un sistema de software.

2.2.7 Indexación de Casos

Las técnicas de indexación garantizan una recuperación eficiente de los casos, entendiéndose por eficiencia tanto los aspectos relacionados con el tiempo de recuperación como la garantía de que ningún caso relevante quede fuera de la búsqueda. Los índices son una estructura de datos que se mantiene en memoria y que puede ser recorrida de forma rápida para evitar la búsqueda directa en el contenido de la LC. Normalmente, los índices estarán constituidos por combinaciones de los descriptores más importantes de los casos para permitir diferenciar unos casos de otros de manera poco costosa [Althoff 2001].

El establecimiento de índices no es una tarea sencilla, sin embargo, es posible utilizar técnicas automáticas de indexación. Los procesos automáticos de indexación han demostrado ser altamente predictivos, siempre y cuando exista un conjunto de características muy bien definidas o cuando se trabaje en dominios reducidos [Juárez+ 2005].

Indexación de casos se refiere a la asignación de los índices de casos para su futura recuperación y la comparación. La elección de los índices es importante porque permite la recuperación del caso correcto en el tiempo correcto [Shiu+ 2004].

2.2.8 Modelo de Memoria de la LC

Se distinguen dos estructuras de almacenamiento [Juárez+ 2005]:

- ***Memoria* Plana**

Esta estructura de almacenamiento presenta los casos de forma secuencial, donde cada elemento de la secuencia es un caso completo (ver Figura 2.4). La ventaja de este tipo de almacenamiento es que la inserción de casos es sencilla, pero la búsqueda de casos es poco eficiente, por lo que siempre sería necesaria la utilización de técnicas de indexación para disminuir el coste que supondría la búsqueda.

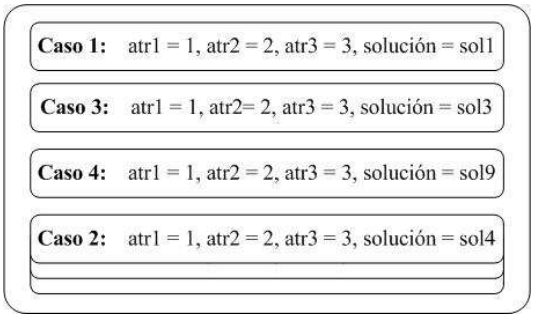


Figura 2.4. Memoria Plana [Juárez+ 2005].

- ***Memoria Jerárquica (MJ)***

En este caso, la estructura de almacenamiento es un grafo, donde cada nodo almacena ciertos atributos compartidos o no por varios casos (ver Figura 2.5). Normalmente, se suele utilizar estructuras de árbol, donde los nodos interiores son atributos comunes y las hojas contienen los atributos diferenciados. Así, el camino desde el nodo raíz a la hoja establece todos los atributos del caso, tanto los comunes como los diferentes, que distinguen un caso de otro. A la hora de establecer la estructura arbórea, se selecciona como nodo raíz aquel atributo más discriminante.

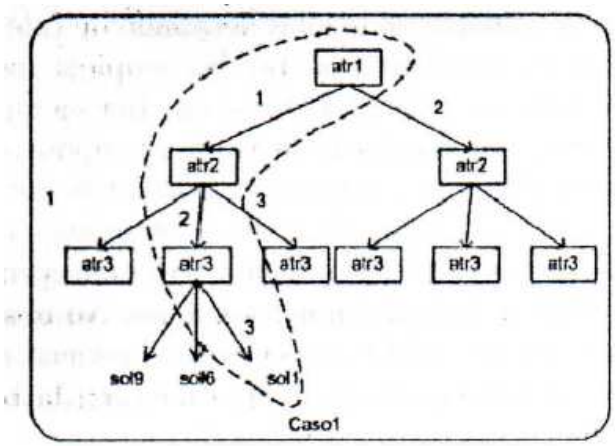


Figura 2.5. Memoria Jerárquica [Juárez+ 2005].

- ***Modelo de Discriminación de Propiedades***

Este modelo mejora el modelo MJ básico en el sentido de que permite la recuperación de casos cuando la descripción del caso a resolver es incompleta. Así, cada nodo de la jerarquía contiene una pregunta acerca de los valores que pueden tomar un atributo, representando los hijos de éstos las respuestas a dichas preguntas. Para que estas jerarquías sean eficientes, los nodos de los primeros niveles hacen referencia a los atributos más discriminatorios, es decir, discriminan antes los casos que no se buscan.

- ***Modelo de Memoria Dinámica***

Es un modelo histórico basado en el concepto de paquetes de organización de modelos o MOPs (del inglés Model Organisation Packages). Uno de los principales problemas del modelo de Discriminación de Propiedades es que el orden de los nodos internos de la jerarquía afecta en la búsqueda del caso. Así, el modelo de Memoria Dinámica organiza los casos en varios grafos (redundante), donde cada uno contiene un orden distinto de los nodos internos (preguntas) [Schank 1982].

Este modelo propone agrupar los casos en unas nuevas estructuras cuando comparten atributos similares denominadas Episodios Generalizados (EG) [Schank 1982].

- ***Episodios Generalizados (EG)***

Los EG están compuestos por normas, índices y casos (ver Figura 2.6). Una norma es el conjunto de todos los atributos comunes de los casos del EG. Un índice establece los atributos discriminatorios y apuntan a un caso o a otro EG más especializado que agrupa otros casos con atributos comunes más específicos. Buscar un caso consiste en ir recorriendo las estructuras y responder a las preguntas a partir de los datos disponibles [Schank 1982].

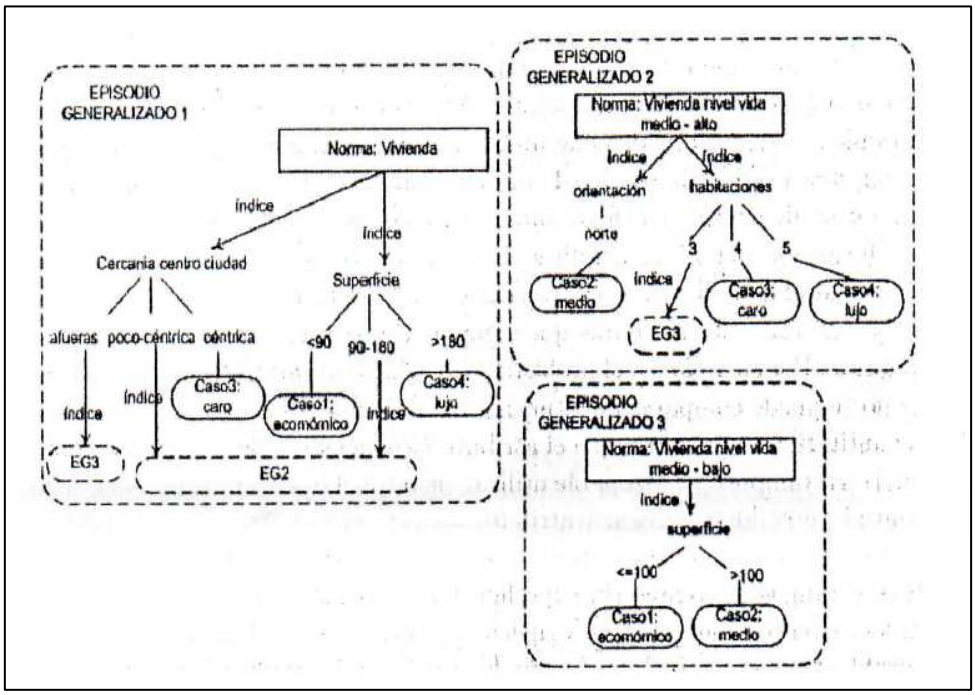


Figura 2.6. Episodio Generalizado [Schank 1982].

- ***Modelo de Categoría y Ejemplo***

En el modelo de Categoría y Ejemplo los casos están distribuidos en una red que representa al mismo tiempo las descripciones de cada caso y el conocimiento del dominio. En este modelo existen tres tipos de elementos: categorías, casos y enlaces. Las categorías agrupan casos similares y los enlaces relacionan los casos con: sus propios atributos, otros casos similares, y otros casos no similares debido a una

característica discriminatoria. Buscar un caso en este modelo consiste en ver qué casos comparten más atributos comunes a partir de los datos disponibles [Bareiss 1998].

2.2.9 Modelo de Ciclo de vida CBR

Este modelo está compuesto por cuatro pasos que engloba un ciclo de razonamiento continuo, buscando en LC problemas similares, adaptando sus soluciones al problema y, finalmente, retroalirnentando la LC con el nuevo problema resuelto. Estas etapas están descritas en un proceso cíclico denominado ciclo CBR, que cubre las cuatro etapas principales del CBR (ver Figura 2.7), denominadas *etapas-RE* [Aamodt+ 1994].

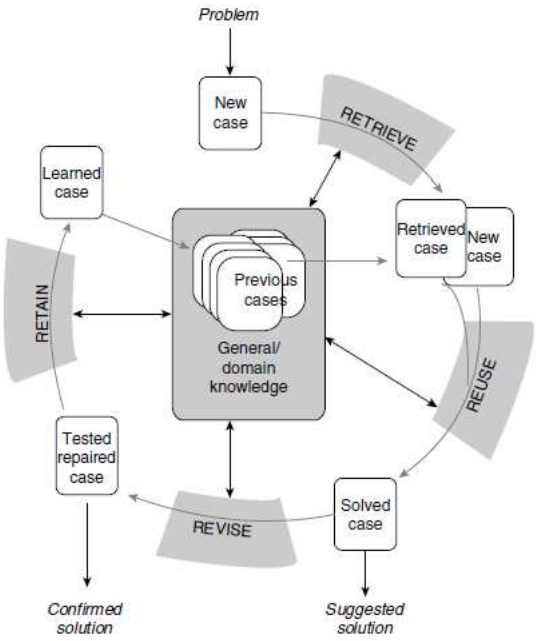


Figura 2.7. Ciclo CBR [Aamodt+ 1994].

Aparte de este ciclo, existe una tarea denominada mantenimiento que, si bien no es parte integrante del proceso CBR, es de vital importancia en el ciclo de vida del SRBC [Althoff 2001].

- ***Recuperar***

El objetivo de esta etapa de **Recuperación** es rescatar de la LC aquellos casos considerados suficientemente similares al problema que se desea resolver, lo cual constituye la entrada de esta etapa. Un paso previo al proceso de búsqueda y una vez que tenemos descrito el problema a resolver, consiste en la identificación de las características relevantes del problema planteado, es decir, selecciona los atributos relevantes mediante algún tipo de filtro y elimina elementos poco descriptivos que pudiesen llevar a búsquedas frustradas [Althoff 2001].

La búsqueda de casos más similares se subdivide en dos pasos:

- En primer lugar, se hace una búsqueda de aquellos casos similares que sean plausibles (este paso es denominado correspondencia inicial). Para obtener estos casos de forma correcta y eficiente, el modo de almacenamiento en la LC (memoria plana y jerárquica) y las técnicas de indexación juegan un papel fundamental.

- Luego de obtener un conjunto de casos candidatos, el segundo paso denominado **selección** consiste en seleccionar de entre los casos candidatos aquellos que sean más similares a la descripción del problema. En esta etapa es necesario establecer cuáles son las medidas de similitud y distancias más adecuadas en el dominio concreto de aplicación. Una de las técnicas más utilizadas para seleccionar los casos más similares es mediante la estrategia de los *k vecinos más cercanos* (kNN), utilizando como función de **evaluación** la función de similitud.

- ***Reutilizar***

Luego de obtener de la LC los casos más similares al problema planteado, el sistema intentará utilizarlos para dar una solución. En algunos casos, bastará simplemente con proporcionar alguna de las soluciones que establecen los casos recuperados. Sin embargo, existen situaciones en las que las soluciones propuestas en los casos recuperados no son aplicables directamente [Althoff 2001].

- ***Revisar***

El objetivo consiste en decidir si la solución propuesta en la etapa anterior es aceptable como solución del problema planteado. Si tras la evaluación de la solución ésta es aceptada, se generará un nuevo caso a partir del par problema-solución, que pasará a la siguiente etapa, Retener [Althoff 2001].

Si la solución no es aceptable, se intentará reparar la solución realizando los cambios pertinentes. Esta etapa es necesaria, ya que la etapa anterior, Reutilizar, puede obtener soluciones incorrectas. De esta manera, se garantiza que ningún caso incorrecto pueda ser incluido en la LC, posibilitando de este modo un proceso de aprendizaje correcto [Althoff 2001].

En la mayoría de las ocasiones esta etapa está supervisada por un experto en el dominio concreto, bien realizando esta tarea de forma manual o semiautomática [Althoff 2001].

- ***Retener***

En esta etapa el objetivo consiste en que una vez que se tiene un caso resuelto y revisado, debe ser incorporado a la LC, terminando de esta manera la fase de aprendizaje. Antes de introducirlo en la LC, debemos decidir qué información, de toda la que proporciona el caso resuelto, se va a incluir [Juárez+ 2005].

En primer lugar, resulta evidente que este nuevo caso tiene que incluir los descriptores del problema (los atributos) y de la solución. Sin embargo, algunos de estos descriptores pueden ser poco o nada relevantes para el conjunto global del caso, por lo que habitualmente se hace una selección de los mismos, este proceso es denominado extracción [Juárez+ 2005].

En segundo lugar, hay que considerar la información obtenida en la etapa de Revisión, por lo que es muy útil incorporar en el caso la meta información relativa a una posible reparación de la solución para aplicarla en un futuro [Juárez+ 2005].

2.2.10 Modelo de descomposición de Tareas-Métodos para CBR

Según [Aamodt+ 1994], con el fin de descomponer y describir los cuatro pasos de alto nivel del ciclo de vida CBR, ellos describen estos procesos como tareas y métodos, donde una tarea particular se realiza aplicando uno o varios métodos.

En la Figura 2.8, las tareas se muestran con los nombres de los nodos en negrita, mientras que los métodos están en cursiva. Los enlaces entre nodos de tarea (líneas continuas) representan varias descomposiciones de la tarea implicada [Aamodt+ 1994].

Por ejemplo, la tarea de más alto nivel que es la Resolución de problemas y aprendizaje mediante la experiencia (problem solving and learning from experience) se descompone en cuatro tareas, que se corresponden con los cuatro pasos del ciclo de vida CBR (ver Figura 2.8): recuperar, reutilizar, revisar y almacenar. Es necesario llevar a cabo las cuatro tareas para poder alcanzar el objetivo de más alto nivel. A su vez, la tarea de recuperación se descompone en las tareas de identificar, buscar, emparejar y seleccionar. Y así sucesivamente [Althoff 2001].

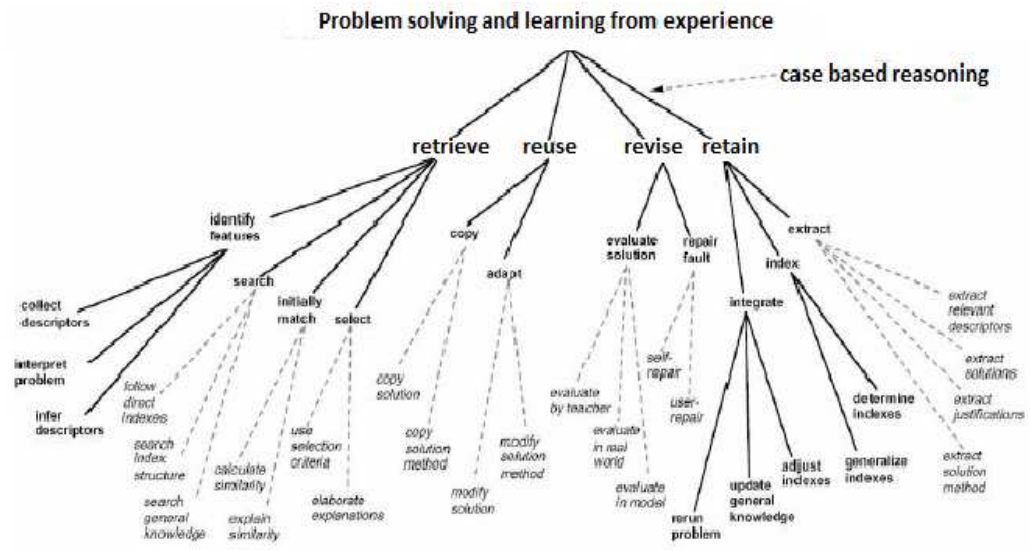


Figura 2.8. Modelo de descomposición de tareas-métodos de CBR [Aamodt + 1994].

Los métodos de cada tarea (líneas discontinuas) indican diferentes formas de llevar a cabo la tarea. Un método especifica un algoritmo que identifica y controla la

ejecución de la subtarea particular, utilizando el conocimiento e información necesarios. Por ejemplo, para la tarea de más alto nivel el método para llevarla a cabo es el razonamiento basado en casos [Aamodt + 1994].

2.2.11 Determinación de casos similares

Luego que se tiene un conjunto de casos, hay que determinar qué caso, o casos, son los más similares y que por lo tanto, se pueden utilizar para resolver el problema planteado. Para llevar a cabo esta fase, resulta imprescindible establecer la manera de medir computacionalmente la similitud entre casos [Althoff 2001].

Así mismo, la propia similitud puede ser interpretada de diferentes maneras: el grado de relación de las características, la situación contextual o la dificultad que supondría su fusión (adaptación). La medida de similitud es el grado de cercanía existente entre los atributos que componen el caso y nos permitirá cuantificar sus correspondencias parciales. De esta manera, las medidas de similitud son las herramientas de las que dispone un Sistema CBR para cuantificar cuan similares son los casos. Así, una medida de similitud contiene conocimiento, más o menos sofisticado, sobre la forma de cuantificar dicha similitud, tradicionalmente entre pares de casos [Juárez+ 2005].

Una forma de definir la similitud entre casos consiste en la determinación de la similitud de los atributos o características del caso. Para ello se debe tener en cuenta que no todos los atributos que componen los casos se pueden comparar de la misma forma [Juárez+ 2005].

- ***Tipos de Similitud***

Una forma de solventar estos problemas consiste en considerar dos medidas de similitud: similitud local y similitud global [Juárez+ 2005].

– **La similitud local** hace referencia a la medida que cuantifica la similitud entre atributos del mismo tipo.

– **La similitud global** mide la similitud para el conjunto de todos los atributos o características de los casos. Se define en función de las similitudes locales.

Para definir las funciones de similitud, vamos a suponer que queremos determinar el grado de similitud entre dos casos, c1 y c2 (ver Figura 2.9), donde todos sus atributos son numéricos, los cuales se representan de la siguiente forma:

$$c_1 = (at_1^1, at_2^1, ..., at_n^1)$$
$$c_2 = (at_1^2, at_2^2, .., at_n^2)$$

Figura 2.9. Grado de Similitud entre dos Casos [Juárez+ 2005].

Para calcular la similitud entre casos (similitud global) se suelen utilizar técnicas basadas en k-vecinos o kNN, para lo que es necesario definir una función de distancia entre los valores de cada atributo [Juárez+ 2005].

Esto es posible ya que existe una relación inversamente proporcional entre el concepto de distancia y la similitud. Es decir, si calculamos la similitud entre dos casos, cuanta menor sea la distancia existente entre los valores sus atributos, mayor es la similitud entre estos. Para ello se calcula la distancia entre los casos combinando las distancias sobre los atributos [Juárez+ 2005].

Distancia Euclidiana

Es una forma sencilla para calcular la similitud (global) entre casos cuyos atributos son cuantitativos (numéricos). Para ello se suelen establecer unos pesos, wk, que determinan la importancia de cada tipo de atributo dentro del caso [Juárez+ 2005].

De esta forma, la similitud local se puede definir corno la distancia entre los casos c1 y c2 (ver Figura 2.10).

$$D(c_i, c_j) = \frac{\sqrt{\sum_{k=1}^{n} w_k d_k(at_k^i, at_k^j)^2}}{\sum_{k=1}^{n} w_k}$$

Figura 2.10. Fórmula del Cálculo de Distancia entre dos atributos [Juárez+ 2005**]**

Donde dk, la función de distancia euclideana entre los valores de los atributos de tipo k y wk, son los pesos que establecen la importancia de cada atributo dentro del caso. En este caso, la similitud global se puede definir en la Figura 2.11 [Juárez+ 2005].

$$S(C_i, C_j) = \frac{1}{D(C_i, C_j)}$$

Figura 2.11. Cálculo de Similitudes entre Casos [Juárez+ 2005].

Por ejemplo, en la Librería de Casos (ver Figura 2.12) se tienen tres casos para una Agencia Inmobiliaria:

Caso1=((2,3,150,'céntrico'),('caro'))
Caso2=((5,1,160,'afueras'),('ecónomico'))
Caso3=((2,1,85,'céntrico'),('asequible'))

Figura 2.12. Librería de Casos, ejemplo [Juárez+ 2005].

Y tenemos como caso de entrada una descripción parcial del apartamento deseado, estableciendo sólo algunos atributos del caso: Cin = ((4,3,-,-), (0))

Donde "—" representa un valor no determinado de un atributo, y el símbolo 0 de la solución indica que es un caso no resuelto. Además, los diferentes atributos del apartamento (dormitorios, aseos, superficie y localización) se han clasificado según su importancia, estableciendo los siguientes pesos (ver Figura 2.13) [Juárez+ 2005]:

$$w_{dormitorios} = 0{,}3$$
$$w_{aseos} = 0{,}05$$
$$w_{superficie} = 0{,}4$$
$$w_{localización} = 0{,}25$$

Figura 2.13. Ejemplo de Asignación de Pesos a atributos [Juárez+ 2005].

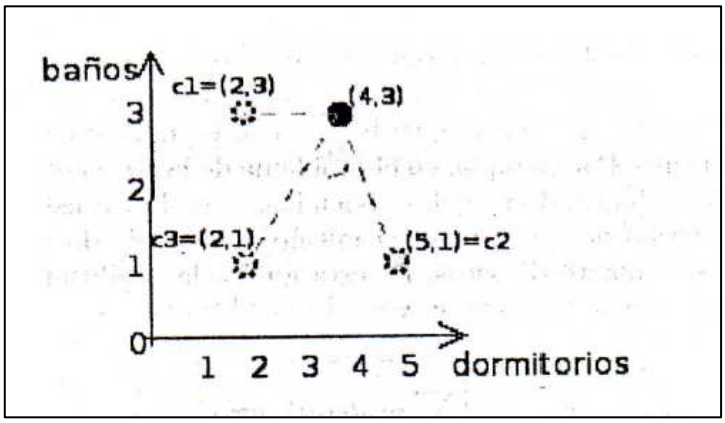


Figura 2.14. Distancia entre atributos [Juárez+ 2005].

El resultado de las distancias entre Cin y los casos almacenados (D(Cin,Ci)) calculadas según las Figuras 2.10 y 2.11 pueden verse en la Tabla 2.1. Como podemos observar, el Caso 2 sería el elegido de entre todos los casos como el más similar [Juárez+ 2005].

Tabla 2.1. Cálculo de las Similitudes, distancia local D y similitud global S para los diferentes casos [Juárez+ 2005].

	Caso 1	**Caso 2**	**Caso 3**
Distancia dormitorio	2	1	2
Distancia Aseos	0	2	2
Distancia Total (D)	**1.7142857**	**1.142857**	**2**
Distancia Global (S)	**0.583333**	**0.875**	**0.5**

En la Figura 2.10 podemos apreciar que, aunque la distancia entre atributos indica la similitud entre los casos (similitud local), los pesos pueden determinar lo contrario respecto a la similitud global. Como podemos observar, una de las mayores dificultades a la hora de describir dicha función radica en la valoración de la importancia de cada atributo (pesos), siendo estos dependientes del dominio de aplicación concreto [Juárez+ 2005].

Distancia Euclidiana Heterogénea

Si observamos el ejemplo desarrollado sobre la similitud, a la medida de similitud propuesta anteriormente se le pueden ver dos problemas.

El primer problema es que no considera aquellos atributos que no están presentes en el caso (por ejemplo el atributo superficie). El segundo, es que está restringido únicamente a atributos numéricos (por ejemplo, el atributo situación no contribuye a la medida de similitud).

Para solventar estos problemas, necesitamos una medida de similitud global mucho más flexible, como la distancia *Euclideana heterogénea*. La distancia *Euclídea heterogénea* básica se define en la Figura 2.15 [Juárez+ 2005].

$$d_{at}(at_i, at_j) = \begin{cases} 1 & si\ at_i \vee at_j\ faltan \\ 0 & si\ at_i \wedge at_j\ nominal\ \wedge\ at_i = at_j \\ 1 & si\ at_i \wedge at_j\ nominal\ \wedge\ at_i \neq at_j \\ \frac{|at_1 - at_2|}{rango_{at}} & si\ at_i \wedge at_j\ lineal \end{cases}$$

Figura 2.15. Fórmula para el cálculo de la *Distancia Euclideana Heterogénea* [Juárez+ 2005].

Esta distancia local calcula un valor de distancia normalizado, es decir, la distancia mínima es cero y la máxima posible es 1. Así, dependiendo de si el atributo es nominal (cadena de caracteres) o numérico, calcula la similitud de forma diversa. Además, penaliza aquellos atributos ausentes dándoles el valor máximo [Juárez+ 2005].

2.2.12 Métodos de adaptación

A continuación describiremos tres de los métodos más utilizados [Juárez+ 2005]:

- ***La reinstanciación***

Es la forma más sencilla de adaptación, puesto que consiste simplemente en copiar la solución del caso recuperado como nueva solución del problema. Se utiliza principalmente cuando el grado de similitud es muy alto entre el caso recuperado y el problema. Su dificultad reside en establecer los umbrales de aceptación del grado de similitud.

- ***La sustitución***

La sustitución de valores consiste en reemplazar valores de partes de la solución de los casos recuperados para obtener la solución del problema. Aquellas partes que contradigan los requisitos del problema se sustituyen, por lo que se requiere de otras técnicas de modelado de conocimiento como la utilización de satisfacción de restricciones, reglas o el ajuste de parámetros. El método de sustitución se suele emplear cuando estamos utilizando una LC sólida.

- ***El método de reparación dirigida por modelos***

Se aplica cuando en los casos recuperados no se encuentra ninguna solución que satisfaga las restricciones del problema planteado. Este método obtiene una nueva solución a partir de las restricciones y características de la solución que se espera obtener. Para ello intenta detectar qué componentes de la solución del caso recuperado se pueden reutilizar, reparando el resto mediante una transformación. Esta transformación es dirigida mediante el conocimiento representado en modelos o heurísticas.

2.2.13 Sistemas CBR

A continuación se presentan algunos sistemas CBR que se han desarrollado:

- ***CHEF [Hammond 1986]***

Es un SRBC cuyo objetivo es la definición de recetas de cocina, entendidas éstas como planes. En CHEF un caso está formado por el conjunto de submetas, los elementos de la receta y la planificación para obtenerlas.

Así, en el ciclo CBR se busca en la LC aquel plan que permita obtener submetas. Para adaptar el plan, en primer lugar se reinstancia la planificación antigua pero sustituyendo los nuevos elementos de la receta, en segundo lugar se realizan labores de ajuste (añadir o eliminar elementos del plan), finalmente, existe un proceso de reparación por si ha habido algún problema con el fin de obtener un plan consistente.

- ***PROTOS [Bareiss 1998]***

Sistema centrado en el diagnóstico de las afecciones del oído. Este sistema intenta clasificar en una de las categorías diagnósticas las evidencias de un paciente encontradas en resultado de pruebas, la historia clínica y los síntomas.

- ***CASEY [Koton 1989]***

Sistema inteligente centrado en el dominio de los fallos cardíacos que combina Razonamiento Basado en Casos y Sistemas Basados en Reglas. Este sistema recupera los casos más similares y determina cuáles son las diferencias entre los casos recuperados con el caso de entrada. Si aparecen pocas diferencias, utiliza un conjunto de operadores de adaptación para obtener un diagnóstico. Si las diferencias son importantes y no es posible la adaptación de la solución, CASEY aplica el módulo basado en reglas para obtener un posible diagnóstico.

- ***BOLERO [López+ 1993]***

Se diseñó para planificar un tratamiento a partir de la información extraída de la historia clínica. Combina un sistema basado en reglas para diagnosticar la enfermedad que presenta el paciente y un componente de CBR para planificar el tratamiento correspondiente.

- ***KRITIK [Goel 1989]***

Se construyó como un sistema híbrido para el diseño de pequeños dispositivos electrónicos y mecánicos, combinando CBR y Razonamiento Basado en Modelos. La parte CBR se utiliza para proponer una solución a partir de diseños ya almacenados y, posteriormente, se utilizan los modelos (conocimiento causal) para comprobar si dicho diseño es válido, o bien si es necesaria la adaptación, en cuyo caso sugiere dónde se debe hacer.

- ***JULIA [Hinrichs+ 1991]***

El objetivo de este SRBC es el diseño de menús para restaurantes. Para ello obtiene el menú más similar y establece el conjunto de restricciones que el nuevo plato debe cumplir. Para obtener la solución utiliza una librería de métodos de adaptación con el fin de conseguir platos que satisfagan las restricciones.

- ***CADET [Sycara 1992]***

Este SRBC fue diseñado para servir como un asistente en el diseño mecánico. El sistema recupera casos pasados que representan diseños exitosos, al mismo tiempo que evita los diseños que condujeron a algún fallo, tales como aquellos que presentaron problemas con la calidad de los materiales o elevaron los costes de producción. CADET transforma una descripción abstracta del comportamiento que se desea en el dispositivo a diseñar, en una descripción que pueda ser utilizada en la recuperación de diseños previos relevantes para el problema actual y, posteriormente, genera un conjunto de diseños alternativos para un conjunto de especificaciones de diseño.

- ***Clavier [Juárez+ 2005]***

Este sistema fue desarrollado por la empresa Lockheed en 1897, fue uno de los primeros sistemas CBR comerciales que estaba orientado al diseño en autoclaves. En la industria aeronáutica se necesita fabricar componentes formados por materiales compuestos que necesitan tratarse simultáneamente a altas temperaturas. Para ello se

utiliza una autoclave, un dispositivo similar a una olla a presión de grandes dimensiones, que permite someter a altas temperaturas lo que se encuentre en su interior (para realizar esterilización o tratamientos térmicos).

- ***FormTool [Juárez+ 2005]***

Este sistema fue desarrollado por la empresa General Electric Plastics en 1994, su objetivo era determinar cuáles son los colorantes que deben ser utilizados para fabricar un plástico de un color determinado.

El problema que resuelve este sistema es encontrar la fórmula que componga un nuevo color para un plástico. Para ello, sabiendo ciertas propiedades del nuevo color, se realiza una búsqueda en un catálogo de muestras y sus composiciones de colores ya utilizados para encontrar el más parecido.

Una vez encontrado, se alterará la fórmula de su composición para obtener el nuevo color. Para realizar este proceso, esta compañía ha utilizado funciones de similitud para seleccionar el color más conveniente, atendiendo a criterios tales como: similitud de color, coste de colorantes y densidad óptica del color.

2.2.14 Herramientas pare el desarrollo de SRBC

Existen varios grupos de investigación que ofrecen herramientas para el desarrollo de SRBC de forma gratuita y que poco a poco van incorporando resultados de investigación. Entre las más destacables podemos citar [Juárez+ 2005]:

- ***FreeCBR [FreeCbr 2011]***

Está diseñado como un motor para CBR y está implementado en Java. Está compuesto por una pequeña herramienta gráfica, para la descripción y búsqueda de casos, un intérprete de comandos, facilidades para desarrollar aplicaciones web y una API de desarrollo.

Los casos se describen mediante un vector de atributos que pueden ser de tipo cadena, multievaluado, entero, real y booleano. Permite configurar la medida de

similitud para la recuperación de casos. Esencialmente, emplea una función de similitud global como la media ponderada de las similitudes locales de los atributos pudiendo seleccionar los pesos.

- ***CBRshell [AIAI 2010]***

Es desarrollado por el Instituto de Aplicaciones de Inteligencia Artificial (AIAI) de Edimburgo, ofrece un shell para CBR que permite la búsqueda de casos más similares y ofrece algunas funcionalidades de optimización del tiempo de ejecución mediante algoritmos genéticos.

Esta herramienta, desarrollado en Java y C++, permite el establecimiento de pesos entre los atributos y configuración de similitudes locales mediante distancias euclídeas (atributos numéricos) y correspondencia de caracteres (atributos de texto). También tiene un módulo que permite configurar los algoritmos genéticos para la optimización.

- ***CBR*Tools [INRIA 2011]***

Desarrollado por el Instituto Francés de Investigación (INRIA), se presenta como un framework orientado a objetos. Este framework facilita el análisis y el desarrollo de SRBC, definiendo en UML los elementos fundamentales de la gestión de un SRBC y realizando la definición de los casos y la organización de la memoria.

- ***jColibrí [GAIA 2011]***

Es un desarrollo del Grupo de Aplicaciones de Inteligencia Artificial (GAIA) de la Universidad Complutense de Madrid. JColibrí también se distribuye como un framework orientado a objetos desarrollado en Java. Una de las principales características de jColibrí es que ofrece una aplicación parcialmente desarrollada y genérica que debe ser extendida e implementada para que sea aplicada a un dominio concreto.

A diferencia de otros frameworks, el desarrollo en jColibrí está enfocado desde el punto de vista de la Ingeniería del Conocimiento, proporcionando una infraestructura

basada en tareas, subtareas y métodos modelados mediante la ontología CBROnto. Un ejemplo de ello es que, para las cuatro etapas del ciclo CBR, jColibrí ofrece una librería de métodos de resolución (PSM). Además, jColibrí permite la configuración de aplicaciones, basada en un motor de inferencia de Lógica Descriptiva, utilizando una interfaz gráfica que permite la selección de métodos para la resolución de las diferentes tareas.

- ***ESTEEM [Stottler 2011]***

Es una herramienta comercializada por Stottle Henke Associates Inc. para el desarrollo de sistemas CBR para la toma de decisiones. Esta herramienta tiene una cómoda interfaz gráfica que permite la definición de casos, la definición de funciones de similitud, la edición de casos y un editor de reglas.

Un caso está definido por un conjunto de atributos, sin embargo, una de las particularidades de ESTEEM es que, aparte de los tipos habituales (booleano, cadena, numérico), permite que existan casos como atributos de otros casos, pudiendo obtener casos anidados. Además, soporta varios métodos para describir la función de similitud local y para la similitud. Adicionalmente, en la LC se pueden establecer reglas IF-THEN, tanto para establecer la similitud como para formalizar el proceso de adaptación de casos.

- ***Art*Eiiterprise [Mindbox 2010]***

Es comercializado por la compañía Mindbox es un entorno de desarrollo de aplicaciones para la toma de decisiones basado en reglas y CBR. Este motor de CBR incluye un entorno que permite desarrollar aplicaciones programando directamente en C++ o en un lenguaje script propio denominado ARTScript.

Así, es posible definir nuevos casos, crear bases de casos, establecer consultas o describir el peso de los atributos. Además, es posible describir estructuras complejas de los casos. Por ejemplo, se pueden definir tipos de atributos como una jerarquía de elementos, atribuyéndole distancia entre ellos para medir similitudes. El entorno

Art*Enterprise también incluye un completo control de versiones de desarrollo y recursos de integración y conectividad con diferentes bases de datos.

- ***ReCall* [*AliceSoft 2010]***

Es una herramienta de desarrollo de aplicaciones CBR y modelado del conocimiento comercializada por la empresa AliceSoft e implementada en C+ +. Permite una representación de casos orientada a objetos utilizando mecanismos de herencia y relaciones entre objetos, ofreciendo una interfaz visual para modelarlo.

Esto permite un modelado del conocimiento muy flexible y potente. Los casos son introducidos en la LC como instancias de estas redes de objetos. Ofrece mecanismos de indexación jerárquicos para realizar recuperaciones eficientes. ReCall permite la generación automática de índices, pero también permite que sean modificados por los desarrolladores.

La similitud entre casos está basada en *knn.* Implementa la etapa de Adaptación mediante dos modalidades: por un proceso de votación, o definido por el usuario mediante un conjunto de reglas.

2.3 Marcos Conceptuales y Glosario

- ***Captación* de Requisitos.** La captura de requisitos es la primera de las actividades a realizar en el proceso de Ingeniería de Requisitos de software y es una de las más exigentes para el grupo de ingenieros, ya que de una buena captura se obtiene un producto final de calidad y de bajos costos en la construcción del mismo [Andriano 2006].

- ***Caso*.** Un *caso* es una situación concreta que describe un problema. Se corresponde con una situación previamente experimentada, capturada y aprendida, de tal forma que puede ser reutilizada en la resolución de problemas futuros [Aamodt+ 1994].

- ***CBR (Razonamiento Basado en Casos).*** El Razonamiento basado en Casos es una técnica de la inteligencia artificial que intenta llegar a la solución de nuevos problemas de forma similar como lo hacen los seres humanos, utilizando la

experiencia acumulada hasta el momento en acontecimientos similares. CBR no depende únicamente del conocimiento general del dominio del problema, sino que utiliza el conocimiento específico de casos pasados para reutilizarlo como solución del nuevo problema [Rossillea+ 1993].

- ***kNN (k vecinos más cercanos).*** Es una técnica de aprendizaje supervisado de regresión, conocida como los k vecinos más cercanos. Esta técnica consiste en estimar el valor de un dato desconocido a partir de las características del dato más próximo, según una medida de similitud o distancia [Juárez+ 2005].

- ***LC (Librería de Casos).*** Es la parte de un SRBC que se encarga del almacenamiento y organización de los casos [Juárez+ 2005].

- **LR.** Librería de Requisitos

- ***Requisito*.** Es una condición o capacidad que debe tener un sistema o un componente de un sistema para satisfacer un contrato, una norma, una especificación u otro documento formal [IEEE 1990].

- ***SR* (Especificación del Requisito.** Es un documento donde se encuentra todas las especificaciones funcionales del requisito.

- ***SRBC (Sistema de Razonamiento Basado en Casos).*** Almacena un conjunto de problemas ya resueltos y recibe como entrada un nuevo problema. El sistema intentará resolver el problema de entrada buscando, de entre los problemas resueltos, el más semejante y adaptando su solución al problema de entrada. De esta manera, podemos construir razonadores que resuelven nuevos problemas mediante la adaptación de soluciones que han sido utilizadas para resolver problemas pasados [Juárez+ 2005].

CAPÍTULO 3

ESTADO DEL ARTE

3.1 Técnicas para la captación de requisitos

A continuación se presenta una breve descripción de las técnicas más utilizadas en la captación de requisitos:

3.1.1 Entrevistas

Las entrevistas son la técnica de captación de requisito más utilizada y son característicamente inevitables en cualquier desarrollo, ya que son una de las formas de comunicación más naturales entre las personas. [Andriano 2006].

En las entrevistas se pueden identificar tres fases: preparación, realización y análisis. En la fase de preparación se estudia el dominio del problema, se selecciona a las personas que se va a entrevistar, se determina el objetivo y contenido de las entrevistas y se planifica las entrevistas. En la segunda fase se realiza la entrevista. Por último, en la tercera fase se debe recapitular para confirmar que no ha habido confusiones en la información recogida, agradecer al entrevistado por su colaboración y citarle para una nueva entrevista si fuera necesario [Piattini+ 1996].

Una vez realizada la entrevista es necesario leer las notas tomadas, pasarlas a limpio, reorganizar la información, contrastarla con otras entrevistas o fuentes de información, etc. Una vez elaborada la información, se puede enviar al entrevistado para confirmar los contenidos. También es importante evaluar la propia entrevista para determinar los aspectos mejorables [Piattini+ 1996].

3.1.2 Encuestas

La encuesta constituye una técnica simple que permite obtener información de una gran cantidad de personas de una manera rápida y eficiente [Andriano 2006].

La posibilidad de repetir las observaciones o preguntas sobre un número de sujetos hace que la encuesta sea una de las técnicas más apropiadas para proveer datos que pueden ser explorados estadísticamente para determinar tendencias, promedios y desvíos o alejamientos de los valores centrales. Las encuestas deben elicitar el punto de vista del cliente, sus metas, procesos, problemas, mejoras sugeridas y estructura lógica del producto, entre otros [Green 2003].

3.1.3 Brainstorming (Tormenta de Ideas)

El brainstorming o tormenta de ideas es una técnica de reuniones en grupo cuyo objetivo es la generación de ideas en un ambiente libre de críticas o juicios [Raghavan+ 1994].

Las sesiones de brainstorming suelen estar formadas por un número de cuatro a diez participantes, uno de los cuales es el jefe de la sesión, encargado más de comenzar la sesión que de controlarla [Raghavan+ 1994].

3.1.4 Casos de Usos

Los casos de usos son una técnica para la especificación de requerimientos funcionales propuesta inicialmente en [Jacobson+ 1993] y que actualmente forma parte de la propuesta de UML [Booch+ 1999].

Un caso de uso es la descripción de una secuencia de interacciones entre el sistema y uno o más actores en la que se considera al sistema como una caja negra [Durán+ 2000].

Los actores son personas u otros sistemas que interactúan con el sistema cuyos requerimientos se están describiendo. Un actor puede participar en varios casos de uso y un caso de uso puede estar relacionado con varios actores [Durán+ 2000].

3.1.5 JAD (Joint Application Design)

La técnica denominada JAD (Joint Application Development), desarrollada por IBM en 1977, es una alternativa a las entrevistas individuales que se desarrolla a lo largo de un conjunto de reuniones en grupo durante un periodo de 2 a 4 días. En estas reuniones se ayuda a los clientes y usuarios a formular problemas y explorar posibles soluciones, involucrándolos y haciéndolos sentirse partícipes del desarrollo [Dube+ 2010].

Esta técnica se basa en cuatro principios: dinámica de grupo, el uso de ayudas visuales para mejorar la comunicación (diagramas, transparencias, multimedia, herramientas CASE, etc.), mantener un proceso organizado y racional, y una filosofía de documentación (lo que se ve es lo que se obtiene), por la que durante las reuniones se trabaja directamente sobre los documentos a generar [Dube+ 2010].

Debido a las necesidades de organización que requiere y a que no suele adaptarse bien a los horarios de trabajo de los clientes y usuarios, esta técnica no suele emplearse con frecuencia, aunque cuando se aplica suele tener buenos resultados, especialmente para capturar requerimientos en el campo de los sistemas de información [Dube+ 2010].

3.1.6 Prototipación

La técnica de prototipación es el proceso de creación de un sistema que ilustra las características relevantes del mismo. Comienza por el estudio preliminar de los requerimientos del usuario y concluye con una serie de requerimientos formulados y un prototipo desechable [Andriano 2006].

Es un proceso iterativo el cual permite la definición del comportamiento del sistema y, en particular, la apariencia del mismo. Esta técnica es realmente útil cuando el prototipo puede ser construido rápidamente [Andriano 2006].

Un prototipo de requerimientos de software es una implementación parcial del sistema, se construye para ayudar a los desarrolladores, usuarios y clientes en la

obtención de un mejor entendimiento del sistema, en especial de los requerimientos que están menos claros [Andriano 2006].

3.1.7 Win Win

Es una técnica usada en la captación de requisitos basada en la recolección a través de una herramienta de soporte. Esta técnica da resultados positivos tanto al desarrollador como al usuario, ya que es iterativa e incremental y ayuda también en la determinación de prioridades [Chao+ 2009].

Las principales características documentadas de la técnica Win Win son: acentúa la necesidad de conseguir un aporte de cada una de las partes implicadas, la aceptación al sistema es una resolución muy probable, se facilita la resolución de conflictos a nivel social, finalmente, la evolución de los requerimientos es activa [Boehm+ 1998].

Para llevar adelante esta técnica se debe realizar los siguientes pasos [Boehm+ 1998]:

- Primero se debe identificar los stakeholders que sean críticos para el éxito del proyecto.
- Luego, identificar las condiciones de ganancia de los stakeholders, las cuales representan los requerimientos específicos de sistema.
- Las condiciones de ganancia no conflictivas y el compromiso de los stakeholders son propuestas como una especificación de sistema.
- Negociar los acuerdos de Win Win de alto nivel, es decir, inventar opciones para la ganancia mutua.
- Agrupar los acuerdos de Win Win.
- Desarrollar los puntos anteriores hasta que el producto esté desarrollado completamente.

3.1.8 Puntos más relevantes de cada Técnica

Luego de ver en detalle cada una de las técnicas más utilizadas para la captación de requisitos, a continuación se mencionan los puntos más relevantes de cada una:

Con las *Entrevistas* se obtiene una gran cantidad de información adecuada a través del usuario, ya que estas suelen ser flexibles.

Las *Encuestas* permiten obtener información de una gran cantidad de *personas* de una manera rápida y esta información puede ser reutilizada en nuevos proyectos.

La técnica Brainstorming o Tormenta de Ideas ayuda a generar una gran variedad de vistas del problema y a formularlo de diferentes formas, sobre todo al comienzo, cuando los requerimientos son todavía muy difusos.

Los Casos de Uso representan los requerimientos desde el punto de vista del usuario, por ello son fácilmente comprensibles. Además, pueden servir de base en las pruebas y documentación del sistema.

Con la técnica JAD, la información obtenida se puede contrastar in situ porque están todos los interesados involucrados, tanto desarrolladores y usuarios finales.

La técnica de Prototipación es útil cuando el prototipo puede ser construido rápidamente.

Usando la técnica Win Win el desarrollo de la captación de requisitos es continuo y colaborativo, donde la refinación de los requisitos se da en cada ciclo.

3.1.9 Análisis de cada Técnica

Después de saber cuáles son los puntos más relevantes de cada técnica, a continuación haremos un análisis de cada una, con respecto al registro o almacenamiento de la información o requisitos que se obtienen.

Podemos notar que la información obtenida con las entrevistas es registrada en los apuntes que realiza el entrevistador. En cambio en las encuestas dicha información se encuentra en los resultados de los cuestionarios.

Usando la técnica Brainstorming, la información obtenida es registrada en alguna plantilla o documento. Mientras que en los Casos de Uso es registrada en los gráficos y notaciones que esta técnica propone.

Con la técnica JAD, los requisitos obtenidos se registran usando alguna herramienta de apoyo: gráficos, documentos, hojas de cálculo, etc., de igual forma que la técnica Win Win. En cambio usando la técnica de Prototipación la información obtenida se refleja a través de los prototipos visuales.

Si nos ponemos a analizar, indirectamente todas estas técnicas están usando Razonamiento Basado en Casos, ya que ellas almacenan la información obtenida y luego la usan para resolver un nuevo problema, sólo que esto no está automatizado. Por ejemplo, los expertos que usan la técnica *Brainstorming* en sus reuniones usan experiencias vividas anteriormente para resolver el problema específico de la reunión.

3.2 Revisión de CBR en la Ingeniería de Software

3.2.1 Uso de CBR en la Optimización de Calidad del Software

Adam Brady y Tim Menzies [Brady+ 2010] realizan una comparación entre los modelos Paramétricos y el Razonamiento Basado en Casos aplicado en la optimización de la Calidad del Software. La optimización de la calidad es un problema no lineal, donde la mejora de cualquier objetivo puede dañar a cualquier parte del proyecto.

En este trabajo los autores usan dos herramientas para realizar la comparación en la optimización de la calidad del software. Las herramientas son las siguientes [Brady+ 2010]:

- ***Herramienta SEESAW***

Es una herramienta que usa un algoritmo que explora los modelos paramétricos de desarrollo de software sobre la base de COCOMO. SEESAW es una herramienta elaborada que sólo puede procesar los datos del proyecto, se expresa en una ontología particular, es decir, sólo los atributos de COCOMO.
Al usar SEESAW los autores encontraron algunos inconvenientes [Brady+ 2010]:

- La dependencia del modelo: SEESAW requiere de un modelo para generar las estimaciones.
- Dependencia de datos: SEESAW sólo puede procesar los datos del proyecto en un formato compatible con el modelo subyacente.
- Diseño arbitraria: SEESAW maneja dos docenas de casos con reglas diseñadas con el "juicio de ingeniería", es decir, no se basan en los resultados teóricos o empíricos (por ejemplo, "no aumentar el uso de herramientas automáticas, sin aumentar la capacidad de analista"). La presencia de tales normas ad hoc hace más difícil comprobar que la herramienta es la correcta.
- Rendimiento: SEESAW utiliza decenas de miles de iteraciones, con un esfuerzo de varios cálculos necesarios calculado para cada iteración. Esto dio lugar a una desventaja de rendimiento.
- Tamaño y mantenimiento: debido a todos los factores mencionados, la base del código SEESAW ha demostrado ser difícil de mantener.

- ***Herramienta W***

Es una herramienta que usa un algoritmo donde aplica Razonamiento Basado en Casos que hace la misma tarea que SEESAW, sin necesidad de utilizar un modelo paramétrico. "W" es fácil de explicar y rápido de construir. No tiene supuestos paramétricos y, por lo tanto, puede ser aplicado rápidamente a datos de los proyectos en diversos formatos. En las pruebas que han realizado "W", se interpretaba estadísticamente mejor, con un nivel de confianza del 95%. Otra ventaja de "W" es que al usar CBR tiene una estrategia de mantenimiento integrado, sólo tiene que añadir más casos.

“W” es una herramienta que usa un algoritmo donde aplica Razonamiento Basado en Casos. CBR en particular se caracteriza [Aamodt+ 1994] en cuatro pasos:

- Recuperar: buscar en los casos más similares al problema objetivo.
- Reutilizar: adaptar nuestras acciones llevadas a cabo en los casos anteriores para resolver el nuevo problema.
- Revisar: revisar la solución propuesta para el nuevo problema y verificar contra la base de caso.
- Retener: conservar las partes de la experiencia actual en la base de caso para la solución de problemas en el futuro.

El procedimiento estándar para CBR es reportar el valor de la clase media de una vecindad local. Esta vecindad es típicamente definido como la distancia euclideana de un proyecto definido en el espacio n-dimensional con n características del proyecto [Brady+ 2010]. “W” funciona de manera similar, pero se define un proyecto como una serie de valores. Finalmente, el objetivo de la “W” se encuentra en el valor más pequeño “i”, de tal manera que “i” selecciona los casos con las mejores estimaciones. En ese punto de vista, la característica distintiva de la CBR es que su razonamiento está basado en instancias y no genera ninguna generalización [Brady+ 2010].

3.2.2 Uso de CBR en el Diseño de Controles para Sistemas de Información basados en Internet (IIS)

Sangjae Lee y Kyoung Kim [Lee+ 2009] presentan una propuesta donde aplican Razonamiento Basado en Casos para el Diseño de Controles para Sistemas de Información basado en Internet (IIS).

En el trabajo que presentan Sangjae Lee y Kyoung Kim sugieren IISCBR (Diseño de los controles para IIS utilizando Razonamiento Basado en Casos), donde un caso modelo se basa en el razonamiento para la generación de recomendaciones de los controles de IIS.

La base de caso de IISCBR consta de ranuras que incluyen entornos de sistemas y controles de IIS. Los Controles de IIS, que son los más demandados en los entornos de sistemas determinados, pueden ser sugeridos por los dos pasos siguientes:

- En primer lugar, el nivel más probable de los controles se sugiere a partir de los casos recuperados.
- En segundo lugar, el nivel de control que tienen los valores más altos en el rendimiento entre el caso recuperado se determina.

Los controles de los IIS se puede recuperar de los casos similares y proporcionar recomendaciones para el control basados en los casos en los últimos IISCBR. Con el fin de evaluar la eficacia de IISCBR, los autores comparan la capacidad predictiva del sistema con el de Análisis Multivariado Discriminante (MDA). Los resultados indican que el uso de Razonamiento Basado en Casos supera a la MDA en la precisión de la predicción.

3.2.3 Uso de CBR para la Estimación del Esfuerzo en el desarrollo de aplicaciones Web Hipermedia.

Emilia Mendes, Ian Watson y Nile Mosley [Mendes+ 2002] presentan una propuesta donde aplican Razonamiento Basado en Casos para estimar el esfuerzo en el desarrollo de aplicaciones Web Hipermedia.

Los autores optaron por utilizar CBR como una técnica de predicción por las siguientes razones:

- Un estudio inicial comparando CBR con las técnicas de predicción de estudios anteriores [Mendes+ 2000], CBR presenta la mejor precisión de la predicción cuando se utiliza un conjunto de datos de proyectos web hipermedia.
- Un segundo estudio, utilizando un conjunto de datos diferentes, también mostró una buena precisión de la predicción usando CBR [Mendes+ 2001].
- CBR es un método intuitivo y hay evidencia de que los expertos aplican el razonamiento analógico al realizar las estimaciones [Hughes 1997].

- CBR es simple y flexible, en comparación con modelos algorítmicos.
- CBR se puede utilizar en datos cualitativos y cuantitativos, lo cual refleja más los tipos de conjuntos de datos que se encuentran en la vida real.

El uso de Razonamiento Basado en Casos implica [Mendes+ 2002]:

- La caracterización de un proyecto de "P" para los cuales se requiere una estimación, es decir, la identificación de los atributos del proyecto (tamaño de la aplicación, la complejidad de enlace, etc.) que pueden influir en el esfuerzo.
- El uso de esta caracterización, como base para encontrar proyectos completos similares, para conocer el esfuerzo.
- El uso de estos valores de esfuerzo, posiblemente con el ajuste, para generar un valor de predicción de los esfuerzos para "P".

- ***Parámetros***

Cuando se utiliza CBR hay una serie de parámetros para decidir, donde cada parámetro, a su vez, puede dividirse en más detalle y puede ser incorporado para una determinada herramienta de CBR, lo que permite varias configuraciones de CBR [Mendes+ 2002]:

- **Medida de Similitud.** Medida de similitud que mide el nivel de similitud entre los proyectos, es decir, de los casos. A nuestro entender, la medida de similitud más utilizado en la ingeniería de software y la literatura de ingeniería Web es la distancia Euclidean sin ponderar. En esta investigación han utilizado tres medidas de similitud: la distancia Euclidean ponderada, la distancia Euclidean sin ponderar y la medida máxima.

  La **Distancia Euclidean sin ponderar**: la distancia Euclidean "d" entre los puntos (x0, y0) y (x1, y1) se ilustra en la Figura 3.1 mediante la representación de coordenadas en un espacio de dos dimensiones:

$$d = \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2}$$

Figura 3.1. Cálculo de la Distancia Euclidean sin ponderar [Mendes+ 2002].

Donde:

d: distancia,
X: atributo X,
Y: atributo Y

El número de atributos empleados determinará el número de dimensiones utilizadas.

Distancia Euclidean Ponderada: es común en CBR para los atributos, es decir, las características de los vectores deben de ser ponderados para reflejar la importancia relativa de cada función. La distancia Euclidean ponderada "d" entre los puntos (x0, y0) y (x1, y1) viene dada por la siguiente fórmula (ver Figura 3.2):

$$d = \sqrt{w_x (x_0 - x_1)^2 + w_y (y_0 - y_1)^2}$$

Figura 3.2. Cálculo de la Distancia Euclidean Ponderada [Mendes+ 2002].

Donde:

d: distancia
X: atributo X
Y: atributo Y
W: peso

Los pesos W_x y W_y son los pesos de X e Y, respectivamente (ver Figura 3.3).

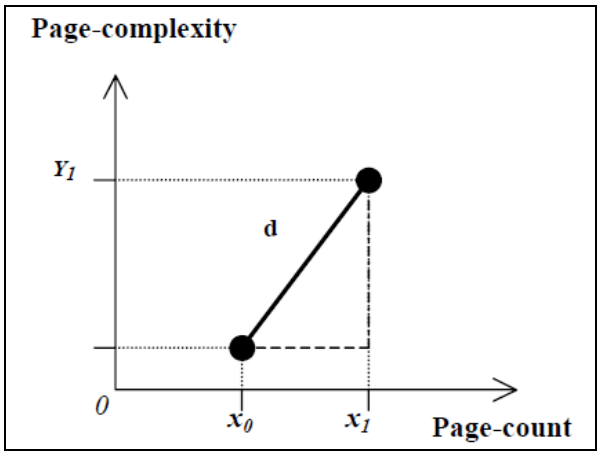


Figura 3.3. Distancia Euclidean ponderada usando dos atributos [Mendes+ 2002].

La **Medida máxima:** usando la medida máxima, la similitud máxima característica define la similitud de casos. Para los dos puntos (x0, y0) y (x1, y1), la máxima medida "d" es equivalente a la fórmula (ver Figura 3.4):

$$d = \sqrt{\max((x_0 - x_1)^2, (y_0 - y_1)^2)}$$

Figura 3.4. Medida Máxima [Mendes+ 2002].

Esto reduce efectivamente la medida de similitud bajo una sola característica, aunque la función de máxima puede ser diferente para cada episodio de recuperación.

- **Escala.** La escala representa la transformación de valores de los atributos de acuerdo a una regla definida, de tal manera que todos los atributos se miden con la misma unidad. Una posible solución consiste en asignar una para el valor máximo observado "m" y luego dividir todos los valores de "m". Esta fue la estrategia elegida para el análisis llevado a cabo utilizando CBR-Works [Schulz 1999].

- **El número de analogías.** El número de analogías se refiere al número de proyectos similares (casos) que se utilizará para generar la estimación. En general, sólo los casos más similares son seleccionados.

– **Adaptación de la Analogía.** Una vez que el caso más similar se ha seleccionado, el próximo paso es decidir cómo generar la estimación. La elección de las técnicas de adaptación de la analogía puede variar en el vecino más cercano, la media de las analogías más cercanas, la mediana, la distancia media ponderada inversa y puntuación media ponderada inversa, por ilustrar unos pocos. La adaptación es la media de las analogías más cercanas. Los autores se decidieron por la media, la mediana y la media ponderada de rango inversa [Mendes+ 2002].

Los autores en este trabajo hacen un análisis basándose en dos conjuntos de datos que contiene información sobre las aplicaciones Web Hipermedia desarrolladas por estudiantes de postgrado de la Universidad de Auckland [Mendes+ 2002].

El primer conjunto de datos (DS1, 34 solicitudes) se obtuvo mediante un estudio de caso (CS1) que consiste en el diseño y la creación, por cada estudiante, de las aplicaciones Web hipermedia dirigido a la enseñanza de un tema elegido, estructurado de acuerdo a la teoría de la flexibilidad cognitiva (CFT) principios, utilizando un mínimo de 50 páginas. Cada aplicación web hipermedia proporciona 46 piezas de los datos, de los cuales se identificaron 8 atributos, que se muestra en la Figura 3.6, para caracterizar a una aplicación web hipermedia y su proceso de desarrollo [Mendes+ 2002].

Metric	Description
Page Count (PaC)	Total number of html or shtml files
Media Count (MeC)	Total number of original media files
Program Count (PRC)	Total number of JavaScript files and Java applets
Reused Media Count (RMC)	Total number of reused/modified media files.
Reused Program Count (RPC)	Total number of reused/modified programs.
Connectivity Density (COD)	Average number of internal links[1] per page.
Total Page Complexity (TPC)	Average number of different types of media per page.
Total Effort (TE)	Effort in person hours to design and author the application

Figura 3.5. Métricas de tamaño y la complejidad de DS1 [Mendes+ 2002].

El segundo conjunto de datos (DS2, 25 solicitudes) se obtuvo con otro estudio de caso (CS2) que consiste en el diseño y la creación, por parejas de alumnos, de las aplicaciones Web hipermedia, estructurado mediante una adaptación del Lenguaje Unificado de Modelado, con un mínimo de 25 páginas. Cada aplicación web hipermedia proporciona 42 piezas de datos, en los cuales identificamos seis atributos, que se muestra en la Figura 3.6, para caracterizar a una aplicación web hipermedia y su proceso de desarrollo [Mendes+ 2002].

Metric	Description
Page Count (PaC)	Total number of html files.
Media Count (MeC)	Total number of original media files.
Program Length (PRL)	Total number of statements used in either Javascript or Cascading Style Sheets.
Connectivity Density (COD)	Average number of links, internal or external, per page.
Total Page Complexity (TPC)	Average number of different types of media per page.
Total Effort (TE)	Effort in person hours to design and author the application

Figura 3.6. Métricas de tamaño y la complejidad de DS2 [Mendes+ 2002].

Durante el proceso de aplicación de CBR los autores configuraron la herramienta CBR en función de cuatro parámetros (medida de similitud, escala, el número de analogías y adaptación de analogía). Por lo tanto, se compararon la precisión de la predicción de las varias estimaciones generadas usando categorías diferentes para un determinado parámetro [Mendes+ 2002].

3.3 Herramientas para la captación de requisitos

Hoy en día existen una variedad de herramientas que facilitan la captación de requisitos en el desarrollo de los proyectos de software, tales como:

3.3.1. REM (Requisite Management)

Es una herramienta gratuita de Gestión de Requisitos diseñada para soportar la fase de definición e ingeniería de requisitos de un proyecto de desarrollo software. Nos

permitirá reflejar la correcta captura de requisitos según lo indicado por el cliente obteniendo un documento ordenado y sin perder flexibilidad a la hora de definir el formato estándar de dicho documento [Durán 2011].

La forma de trabajo de REM consiste en la elaboración de un documento con los requisitos que queramos introducir. El documento a generar podemos indicar que sea de varios tipos: documento de requisitos del sistema, documento de análisis del sistema, registro de conflictos y defectos, registro de peticiones de cambio [Durán 2011].

3.3.2. DOOR

Es un producto de IBM Rational, es una herramienta para la gestión de requisitos que puede ayudar a reducir costes, aumentar la eficiencia y mejorar la calidad, ya que permite optimizar la comunicación, la colaboración y la verificación de requisitos en la organización y en toda la cadena de suministros. Esta herramienta considera los requisitos como objetos y los documentos como módulos. Tiene una orientación basada en objetos, frente a RequisitePro y Caliber-RM, que manejan solamente requisitos y sus atributos [IBM-DOORS 2011].

3.3.3. Rational Requirements Composer

Esta herramienta es creada por IBM, permite a los equipos definir, gestionar y presentar requisitos en un proyecto de desarrollo del ciclo de vida. Es una herramienta basada en la web que da soporte a metodologías de desarrollo iterativas, en cascada y fácilmente escalables mediante procesos de requisitos ligeros. Rational Requirements Composer ayuda a gestionar el ciclo de vida de la información de los requisitos, mediante la implementación de un enfoque amplio y flexible que permita a los equipos colaborar, clarificar y llegar a un acuerdo sobre los requisitos [IBM-RRC 2012].

3.3.4. Evaluación de Herramientas

En la propuesta SIREN [Lasheras+ 2011] se basa en la utilización de un modelo de proceso en espiral, unas plantillas de documentos de requisitos y un repositorio de requisitos reutilizables, que se encuentra organizado por catálogos.

Los catálogos de requisitos de SIREN se corresponden con lo que denominamos perfiles (o dominios de aplicación "horizontales", como por ejemplo seguridad, LOPD, etc.) y dominios (dominios de aplicación "verticales", tales como: seguros, banca, etc.). Estos catálogos se componen de una jerarquía de documentos de especificación, estructurados de acuerdo a estándares del IEEE [Lasheras+ 2011].

Todo requisito en SIREN tiene un conjunto mínimo, común, de atributos [Lasheras+ 2011], aunque dependiendo del catálogo puede tener definidos atributos adicionales. Uno de los atributos de los requisitos son las relaciones de trazabilidad exclusiva, que implican que sólo uno de los requisitos que mantengan esta relación podrá ser incluido en un mismo proyecto.

- ***Modelo de proceso de SIREN***

El modelo de proceso de SIREN es básicamente una adaptación del modelo de proceso en espiral propuesto en [Kotonya + 2001], donde los autores han añadido las siguientes actividades (ver Figura 3.8):

- **Selección de Requisitos**. Se proporcionan las plantillas de los documentos de especificación rellenas con requisitos del repositorio. En los dominios y perfiles relevantes para el proyecto actual, el analista junto con el resto de stakeholders seleccionarán los requisitos del repositorio que sean adecuados para la aplicación concreta que se está desarrollando. Nótese que cuando se reutiliza un requisito también se seleccionarán todos los requisitos relacionados con él mediante dependencias inclusivas. Además, se deberá asegurar que no se incluya en la especificación dos requisitos que sean exclusivos.

- **Obtención de requisitos específicos**. De las reuniones entre los analistas y el resto de stakeholders se obtienen los requisitos informales y específicos del sistema actual. Estos requisitos no están inicialmente en el repositorio.

- **Mejora del Repositorio**. El repositorio no se debe considerar como un producto final, estático, sino como un producto en continua evolución, incluyendo nuevos requisitos reutilizables y mejorando la calidad de los existentes.

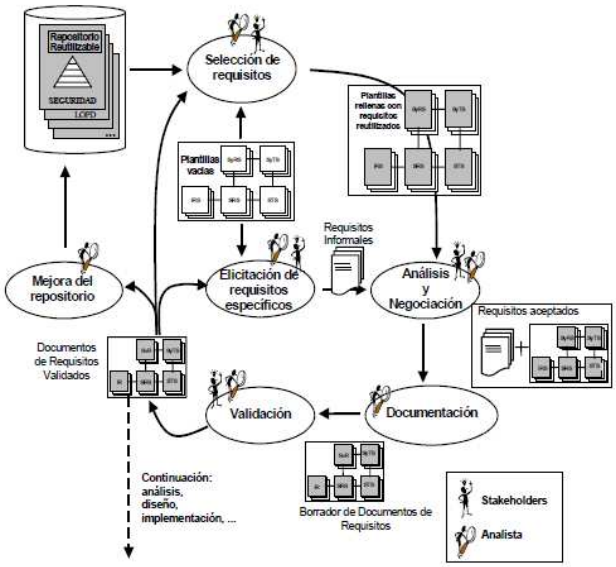


Figura 3.7. Modelo de proceso de SIREN para la reutilización de requisitos [Lasheras+ 2011].

- ***Análisis de Herramientas CARE***

Las herramientas CARE (Computer-Aided Requirements Engineering) dan soporte aplicando los métodos y técnicas de Ingeniería de Requisitos. Estas herramientas permiten la definición, organización y gestión de requisitos, facilitando la puesta en práctica del proceso de Ingeniería de Requisitos, o simplemente haciéndolo posible [Lasheras+ 2011]. Sin embargo, los autores han detectado que una de las actividades más efectivas de la Ingeniería de Requisitos, la reutilización de requisitos, no está contemplada de manera explícita en la mayoría de herramientas CARE actuales.

Los autores [Lasheras+ 2011] analizan la situación actual con respecto a las herramientas CARE y la reutilización. Por ello establecen un marco para realizar una

comparativa entre herramientas. Para definir este marco ellos tuvieron en cuenta dos aspectos: las necesidades generales de una herramienta de gestión de requisitos y las necesidades específicas de reutilización y del método SIREN.

Tras una primera selección los autores eligieron para la comparativa las herramientas Caliber-RM v. 4.0 [Borland 2011], DOORS v. 5.2 [IBM-DOORS 2011] y RequisitePro v. 7.0 [Rational 2011].

- ***Resumen Comparativo entre las Herramientas***

A continuación se muestra un resumen de las características de cada herramienta expuesta anteriormente (ver Figura 3.9).

CARACTERISTICAS NECESARIAS	HERRAMIENTAS		
	CALIBER-RM V.4	DOORS V.5.2	REQUISITEPRO V2002
Importación de Requisitos	Sí, Word y ficheros delimitados.	Sí, Word y ficheros delimitados.	Sí, Word y CSV.
Clasificación de Requisitos	Sí, se definen tipos de requisitos	Sí, se definen objetos (requisitos como objetos).	Sí, se definen tipos de requisitos
Requisitos parametrizados	No	No	No
Soporte selección y Reutilización de Requisitos	**No**. Sólo copiar y pegar dentro y entre proyectos.	**No**. Sólo copiar y pegar módulos.	**No**, sólo copiar y pegar dentro del mismo proyecto.
Asignación de propiedades de rendimiento, etc.	Sí, atributos asociados a tipos de requisitos.	Sí, por medio de atributos asociados a objetos.	Sí, por medio de atributos asociados a tipos de requisitos.
Creación de nuevos tipos de requisitos y atributos	Sí	a nivel de módulos	Sí
Identificación de inconsistencias	Sí, de forma gráfica con matriz de trazabilidad.	Sí, de forma gráfica con matriz de trazabilidad.	Sí, de forma gráfica con matriz de trazabilidad.
Visibilidad de enlaces de trazabilidad	Sí, de forma gráfica. Por tipos de requisitos y para un requisito sólo.	Sí, de forma gráfica, con todos los requisitos.	Sí, de forma gráfica, por tipos de requisitos

Figura 3.8. Resumen Comparativo de tres herramientas [Lasheras+ 2011].

RequisitePro permite la reutilización utilizando plantillas de documentos, al igual que la herramienta REM [Durán 2011], que soporta reutilización utilizando cualquier documento como “documento base”. Sin embargo, en SIREN los autores buscan una reutilización con distintos grados de granularidad, permitiendo al usuario desde reutilizar un solo requisito procedente de alguno de los catálogos disponibles a reutilizar catálogos completos [Lasheras+ 2011].

- ***La Herramienta SirenTool***

Después del estudio de algunas herramientas comerciales, los autores elaboraron la especificación de requisitos (SRS) de la herramienta CARE de soporte a SIREN [Lasheras+ 2011]. A partir del SRS desarrollaron un prototipo de SirenTool, que es el nombre que le dieron a la solución completa, como resultado de la integración de las necesidades propias del proceso SIREN en la herramienta RequisitePro.

Las funcionalidades añadidas están disponibles a través de una interfaz programada con VisualBasic a la que denominamos “SIRENA” (SIREN Automated). En RequisitePro esta interfaz aparecerá como una opción nueva en el menú Tools.

Esta herramienta aporta a la reutilización de requisitos la gestión de requisitos parametrizados, la gestión de las relaciones de traza inclusiva y exclusiva, y añade facilidades de búsqueda más avanzadas a las ya soportadas por las herramientas comerciales. Pero este prototipo de la herramienta se encuentra en continua evolución. Los autores están trabajando en la mejora de sus funcionalidades, ampliando los criterios de búsqueda para la reutilización de requisitos, definiendo nuevos catálogos de requisitos y estudiando la integración con otras herramientas para la definición de los requisitos funcionales [Lasheras+ 2011].

A mi parecer, *podría usarse alguna técnica inteligente que ayude en la reutilización de requisitos en esta herramienta. Como por ejemplo: Razonamiento Basado en Casos.*

CAPÍTULO 4

MODELO PROPUESTO

El siguiente modelo propuesto que se presenta es para la Captación de Requisitos en el desarrollo de proyecto de software haciendo uso de Razonamiento Basado en Casos, el cual constituye el aporte teórico; la justificación del uso de CBR, el mapa del modelo, y la descripción general y específica de cada uno de sus componentes son parte del presente capítulo.

4.1 Justificación del modelo

El presente modelo se sustenta fundamentalmente en los siguientes puntos:

- Según el reporte del caos del año 1995 [StandishGroup 1995], no todos los proyectos de software son terminados con éxito. Sólo el 16 % de los proyectos son terminados con éxito, el 53% son terminados con problemas y el 31 % de estos son cancelados. Existen muchos factores de los fracasos de estos proyectos, uno de ellos se encuentra en los *Requerimientos* (ver Figura 1.1 y Tabla 1.2).

- Para mitigar este problema existen muchas herramientas y técnicas que facilitan el trabajo en la administración de requisitos, tales como: *Brainstorming* [Raghavan+ 1994], *Entrevistas* [Piattini+ 1996], *Casos de Usos* [Booch+ 1999], *Encuestas* [Andriano 2006], *Prototipación* [Andriano 2006], *Win Win* [Chao+ 2009] y *JAD* [Dube+ 2010]. Sin embargo, a pesar de ello siguen los problemas [StandishGroup 2009].

- El Razonamiento Basado en Casos (CBR) es una técnica de Inteligencia Artificial que permite aprovechar la experiencia acumulada en la solución de nuevos problemas. Con esta técnica se almacenan casos con la solución que se ha

dado anteriormente y cuando se presenta un nuevo problema esta información o experiencia acumulada es empleada para resolverlo [Althoff 2001].

- CBR se han aplicado con éxito en la Ingeniería de Software, como por ejemplo: en la Estimación de costos de proyectos de software utilizando la herramienta CBR-Works [Mendes+ 2002], en el Diseño de Controles de Sistemas de Información basados en Internet [Lee+ 2009], en la Optimización de la calidad del software [Brady+ 2010], etc.

- Todavía no se ha aplicado CBR en la Captación de requisitos, es por ello que en el presente modelo se propone usar *CBR* en la *captación de requisitos,* ya que con esta técnica de inteligencia artificial se aprovecha los requisitos de proyectos de software desarrollados anteriormente, para luego resolver o identificar requisitos de un nuevo proyecto.

4.2 Mapa del Modelo

A continuación se presenta el siguiente modelo para la Captación de Requisitos en el desarrollo de proyectos de software aplicando Razonamiento Basado en Casos (ver Figura 4.1).

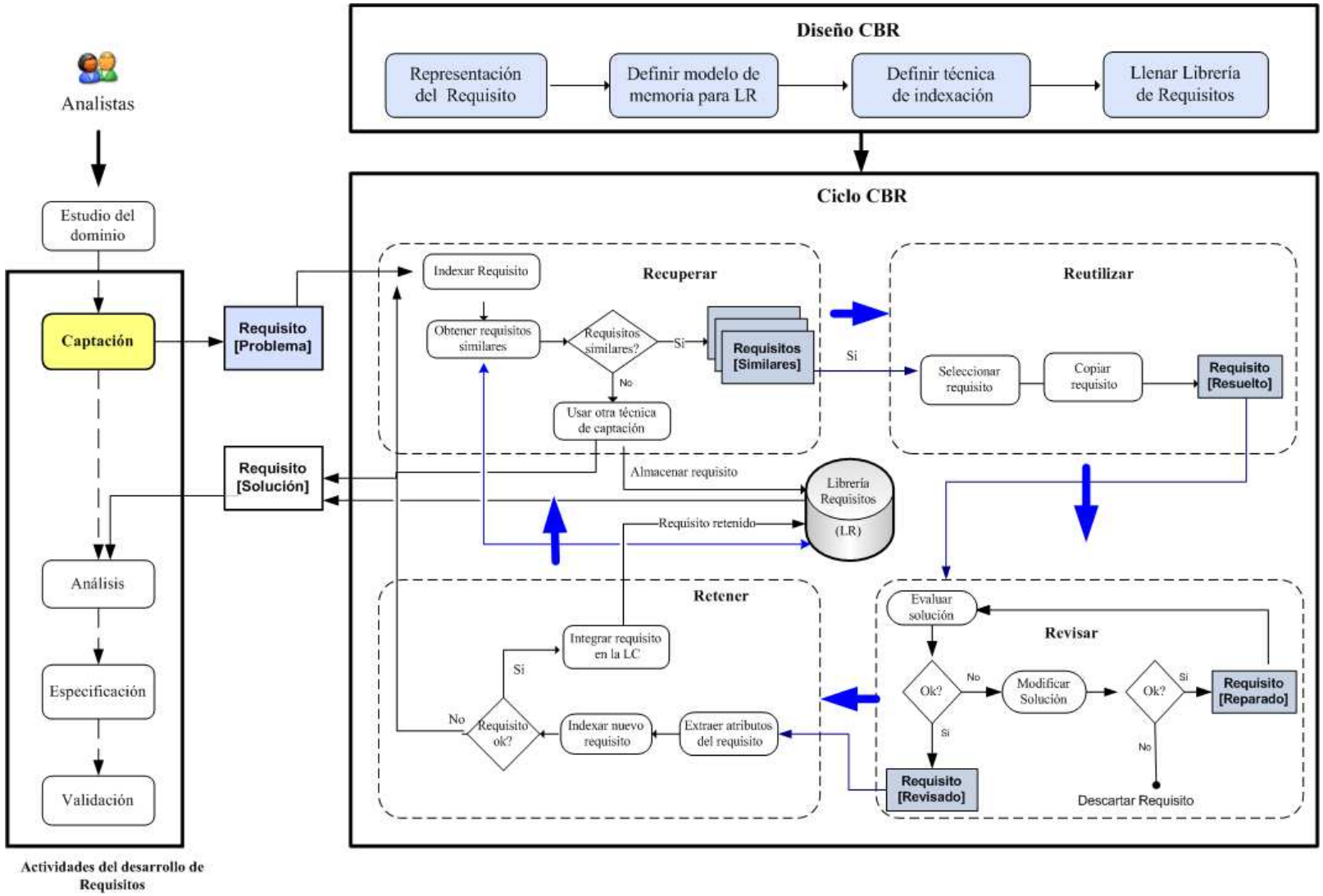


Figura 4.1. Modelo propuesto para la Captación de Requisitos en el desarrollo de proyectos de Software aplicando CBR

4.3 Descripción general del modelo

El presente modelo se sustenta fundamentalmente en los siguientes puntos:

- El modelo que se propone está compuesto por dos etapas: Diseño CBR y Ciclo CBR, tal como se muestra en la Figura 4.1.

- En el estudio del dominio, el analista deberá hacer un estudio del negocio según el tipo de proyecto que va desarrollar. Se propone usar las notaciones gráficas BPMN para definir los diagramas de flujo de los procesos de negocio.

- Tal como se muestra en la Figura 4.1, las actividades del desarrollo de requisitos son: captación, análisis, especificación y validación. Siendo la primera el centro de atención del modelo propuesto.

- En la Captación, el analista deberá identificar el requisito, este requisito o caso es el problema que el analista debe resolver. En el modelo que se propone, la captación de requisitos se realizará usando la técnica de Razonamiento Basado en Casos si se encuentran algún requisito similar en la Librería de Casos, en caso contrario se propone usar RUP.

- En el Diseño CBR, se realizarán todas las tareas relacionadas con la construcción de CBR, tales como: representar el requisito, definir el modelo de memoria para la Librería de Requisitos, definir la técnica de indexación y llenar la LR (Librería de Requisitos) con requisitos de anteriores proyectos.

- En el ciclo CBR, se realizarán todas las tareas y técnicas involucradas en cada uno de los procesos del ciclo de vida de CBR. Para ello se tomará como base el modelo de procesos del *ciclo de vida CBR*, propuesto por [Aamodt+ 1994], donde se usan las cuatro etapas: Recuperar, Reutilizar, Revisar y Retener, para darle solución al nuevo caso, es decir, al nuevo *Requisito*.

- Se utilizará también el modelo de descomposición Tareas-Métodos, propuesto por [Aamodt+ 1994], donde no se utilizan todas las tareas y métodos de éste, sino los más adecuados para nuestro modelo.

- La representación del Requisito estará compuesta por las características o atributos del problema y la solución, según [Juárez+ 2005]. Es decir, este Requisito estará compuesto por atributos donde uno de estos atributos será la solución del problema (ver Figura 4.3).

- Para la selección de Requisitos similares, en el modelo se usará la fórmula de la Distancia Euclideana Heterogénea (ponderada) explicado en la sección 2.2.11 del capítulo 2.

- Para el modo de almacenamiento de la Librería de Requisitos se usará la memoria plana, ya que este tipo de estructura presenta los Requisitos de manera secuencial como arrays, lo que permite que la inserción de Requisitos sea sencilla. Para hacer eficiente la búsqueda de Requisitos se usarán índices [Juárez+ 2005].

- Para la indexación de los Requisitos se propone usar el algoritmo que se muestra en la Tabla 4.2, donde se usará como índices los atributos: *dominio* y *área*, donde se clasifican los *Requisitos* en categorías de acuerdo a estos índices (ver Figura 4.4).

- En la etapa Recuperar del ciclo de vida CBR, se propone usar el algoritmo que se muestra en la Tabla 4.3 Recuperar_requisito, a la vez este algoritmo usa el algoritmo Indexar_requisito para indexar los Requisitos, luego se usa la fórmula de la Distancia Euclideana Heterogenea (ponderada) para calcular la similitud entre los Requisitos, se ordenan los Requisitos similares y se escogen los que tienen mayor grado de similitud.

- En la etapa Reutilizar del ciclo de vida CBR, se propone usar el algoritmo que se muestra en la Tabla 4.4 Reutilizar _requisito, dependiendo del grado de similitud de los requisitos éstos se copiarán.

- En la etapa Revisar del ciclo de vida CBR, se propone usar el algoritmo que se muestra en la Tabla 4.5 Revisar _requisito. En esta etapa se revisa si la solución propuesta para nuestro problema es la adecuada, acá su validez debe ser probada con la realidad.

– En la etapa Retener del ciclo de vida CBR, se propone usar el algoritmo que se muestra en la Tabla 4.6 Retener_requisito. Acá se decide qué información de todo lo que proporciona el Requisitos resuelto se va incluir en nuestro problema, para luego analizar nuevamente si la solución de nuestro caso se incorporará a la Librería de Requisitos.

4.4 Descripción específica del modelo

A continuación se efectúa una descripción detallada de cada una de las partes del modelo propuesto, empezando con las tareas relacionadas con el estudio del dominio, *Diseño de CBR*, y culminando con las tareas realizadas en cada uno de los proceso del *ciclo de vida de CBR* (ver Figura 4.1).

4.4.1 Estudio del dominio

En el estudio del dominio, el analista deberá hacer un estudio del negocio según el tipo de proyecto que va desarrollar. Se propone usar BPMN para definir los procesos de negocio. BPMN es una notación gráfica estandarizada basada en diagramas de flujo para definir procesos de negocio. Modela tanto la secuencia de actividades como los datos o mensajes intercambiados entre los distintos participantes de un proceso [Rational 1998].

4.4.2 Captación de Requisitos

Tal como se muestra en la Figura 2.1, las actividades del desarrollo de requisitos son: captación, análisis, especificación y validación. Siendo la primera el centro de atención del modelo propuesto.

Según la Figura 4.1, en la Captación el analista deberá identificar el requisito, este requisito o caso es el problema que el analista debe resolver. En el modelo que se propone, la captación de requisitos se realizará usando la técnica de *Razonamiento Basado en Casos* si se encuentran algún requisito igual o similar en la Librería de Requisitos, de lo contrario se deberá usar otra técnica para la captación de requisitos del nuevo caso y luego deberá ser registrado en Librería de Requisitos para su futuro uso.

4.4.3 Diseño de CBR

En el Diseño CBR se realizarán todas las tareas relacionadas con la construcción de CBR, tales como: representar el Requisito, definir el modelo de memoria para la Librería de Requisitos, definir la técnica de indexación, crear la Librería de requisitos y llenar la Librería de Requisitos con requisitos de anteriores proyectos (ver Figura 4.2):

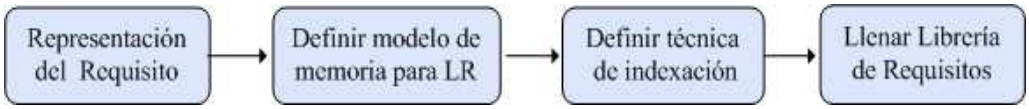


Figura 4.2. Tareas en el diseño CBR.

- ***Representación del Requisito***

En el modelo que se propone, para la representación de los Requisitos se tomarán en cuenta los siguientes puntos:

– El caso será: "Requisito". Se entiende a este caso como un requisito funcional del sistema.

– La representación del Requisito estará compuesta por las características o atributos del problema y la solución, donde la solución del problema se encuentra en el atributo *n* (ver secciones 2.2.2 y 2.26).

Caso = (atributo1, atributo2,... atributo n)

Figura 4.3. Representación del Requisito.

– Un Requisito nuevo será representado por r'.

– Un Requisito que se encuentre en la LR será representado por r.

– Las características o atributos del Requisito son por ejemplo: nombre del proyecto, dominio del sistema, área de la empresa, verbo del requisito,

sustantivo del requisito, actor principal, actor secundario, tipo de principal, tipo de entrada secundaria y Especificación del Requisito (SR). Siendo este último la solución del problema (ver Figura 4.4).

- El SR (Especificación del Requisito), es un documento donde se encuentra todas las especificaciones funcionales del requisito.

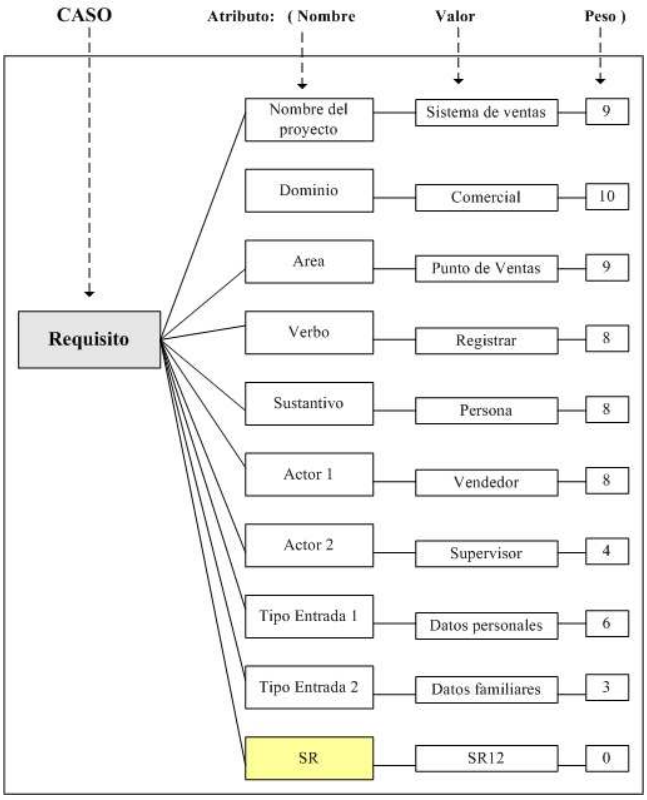


Figura 4.4. Estructura del Requisito.

Por ejemplo, si nuestro caso es un *requisito* del sistema para registrar datos de una persona, los atributos del requisito serían: Sistema de Ventas, Comercial, Punto de ventas, Registrar, Persona, Vendedor, Supervisor, Datos personales, Datos familiares y SR12 (nombre del documento Especificación del Requisito).

- También los atributos del *requisito* está compuesto por *pesos*, estos pesos ayudarán a determinar la importancia de cada atributo. La Figura 4.4 muestra el rango de valores de los pesos de acuerdo al nivel de importancia.

– Para la determinación de los pesos se ha desarrollado un rango de valores, de acuerdo al nivel de importancia de cada atributo (ver Tabla 4.1).

Tabla 4.1. Rango de valores para la determinación de los pesos.

Nivel	Rango de valores
Alto	7 - 9
Medio	4 - 6
Bajo	1 - 3
Nulo	0

- ***Definición del modo de memoria de la LR***

En el presente modelo se propone usar la memoria plana (ver Figura 2.4), ya que este tipo de estructura presenta los Requisitos de manera secuencial como arrays, lo que permite que la inserción de Requisitos sea sencilla. Para hacer eficiente la búsqueda de Requisitos se usarán índices (ver Figura 4.5) [Juárez+ 2005].

- ***Llenar Librería de Requisitos***

Para usar el modelo propuesto, se deberá llenar la Librería de Requisitos con requisitos de proyectos de software que se encuentran ejecutados con éxito. Sólo se podrá ejecutar este modelo, si y sólo si, la LR contiene requisitos de diversos proyectos de software.

- ***Indexación de Casos***

El establecimiento de índices no es una tarea sencilla, es posible utilizar técnicas automáticas de indexación [Juárez+ 2005].

En esta investigación se propone un algoritmo que permitirá indexar los casos, este algoritmo se usará en el proceso de Recuperación.

Para tal tarea se organizan jerárquicamente los casos de manera que sólo un pequeño subconjunto de ellos tiene que ser buscado durante la recuperación. Los puntos que se deben tomar en cuenta son los siguientes:

- Seleccionar el atributo más importante como índice: “Dominio”. Este atributo es considerado como importante porque los requisitos van a ser diferentes de acuerdo al tipo de sistema. Por ejemplo, un requisito de un sistema comercial no será igual a un requisito de un sistema bancario.

- Clasificar los casos en clases o categorías de acuerdo a este atributo: “Dominio”. Donde las categorías de casos se denotarán de la siguiente manera:

  CC = { C1,C2,C3,C4,C5…} (ver Figura 4.5).

- La entrada de un caso nuevo (*requisito*), estará compuesta por un conjunto de atributos denotadas de la siguiente manera: r’ = { f1,f2,f3,f4,f5,f6,f7…f10}

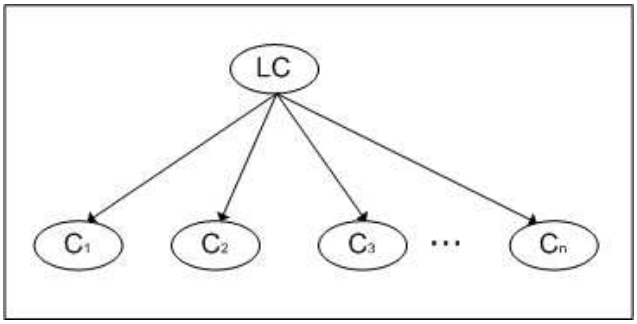


Figura 4.5. Estructura de indexación de la Librería de Casos.

A continuación se plantea el algoritmo que realiza todas las tareas expuestas anteriormente (ver Tabla 4.2):

Tabla 4.2. Algoritmo para indexar un requisito.

Indexar _ Requisito
<u>*Inicio*</u>
Entrada: r’ = {f1, f2, f3, f4, f5, f6, f7,…f10} //Caso nuevo: Requisito
Salida: Ci // Categoría de casos
1. Desde i = 1, seleccionar un caso *rj* al azar de la categoría *Ci*
2. Comparar el valor de la característica “Dominio” de *r’* y *rj*
3. Si los dos valores son idénticos
   Entonces r’∈ Ci,
   Si no i = i+1, ir al paso 1
4. Repetir pasos 1, 2 y 3 hasta que r’ pertenezca a algunas categorías Ci.

<u>*Fin*</u>

4.4.4 Ciclo de vida CBR

A continuación se explican las tareas de cada uno de los procesos:

- ***Recuperar***

Tal como se muestra en la Figura 4.1, este proceso comienza cuando hay como entrada un problema, es decir, un *requisito* r' (caso nuevo), por lo que se plantea realizar las siguientes tareas:

- Indexar Requisito. Para poder recupera los casos o requisitos similares a nuestro problema r', primero se debe indexar los casos, es por ello que se propone usar el algoritmo Indexar_Requisito que se muestra en la Tabla 4.2.

- Usar la técnica de Similitud. Acá se va usar la medida de similitud para seleccionar los casos similares a nuestro problema r'. Se utilizará la fórmula de la Distancia Euclidean Heterogenea (ponderada) (ver sección 2.2.11).

- Obtener Requisitos similares. Acá se escogerá los casos más similares de acuerdo al grado de similitud, es decir, los casos que tengan mayor grado de similitud. El número de casos a seleccionar será a criterio del analista o usuario del sistema CBR.

- Si no existen requisitos similares, es decir, el caso es nuevo, se debe usar otra técnica de captación (podrían ser Casos de Usos, Win Win, etc.).

A continuación se plantea el algoritmo que realiza todas las tareas expuestas anteriormente (ver Tabla 4.3):

Tabla 4.3. Algoritmo para recuperar un requisito.

Recuperar _ Requisito
Inicio

//Entrada: r' = {f1, f2, f3, f4, f5, f6, f7,..f10} Requisito o caso nuevo
//Salida: CS: Conjunto de Casos similares

1. Ejecutar el algoritmo de la Tabla 1 para tener una categoría de casos CC
   // Calcular el grado de similitud entre *r' y rj*
2. Para j=1 hasta k
   SIM(r', rj)
   //Donde:
   // i: contador
   // k: es el número de casos que hay en la clase de casos CC
   // SIM: es la medida de similitud
3. Si existen casos Similares
   – Ordenar los casos de acuerdo a la medida de similitud (SIM) calculados en el paso 2.
   – Escoger los casos más similares de acuerdo al grado de similitud.
4. Si no existen casos similares
   – Usar otra técnica de captación de requisitos.

Fin

- ***Reutilizar***

Tal como se muestra en la Figura 4.1, luego de obtener de la LR los Requisitos más similares al problema r' (CS: Conjunto de casos similares a r') se plantea realizar las siguientes tareas:

– Seleccionar Requisito. Acá se selecciona el requisito con el grado de similitud alta.

– Copiar Requisito. Luego de tener el requisito con el grado de similitud alta se usa la información de éste como parte de la solución de r'. Teniendo así el requisito resuelto para ser analizado en la siguiente etapa. Por ejemplo, si la solución del requisito *ri* es SR20 (nombre del documento SR), entonces se copiará este valor como solución del requisito *r'*. Luego este documento será evaluado en la siguiente etapa.

A continuación se plantea el algoritmo que realiza todas las tareas expuestas anteriormente (ver Tabla 4.4):

Tabla 4.4. Algoritmo para reutilizar requisitos.

Reutilizar _ Requisito
Inicio

// Entrada: CS = r1 ,r2, r3... rk Conjunto de Casos similares
// Salida: ri requisito resuelto

1. Seleccionar el requisito *ri* con grado de similitud alta
2. Copiar la solución del requisito ri a r'
3. Ir a la siguiente etapa

Fin

- ***Revisar***

Tal como se muestra en la Figura 4.1, luego de tener la solución propuesta o requisito resuelto para nuestro problema, es muy importante saber si la solución propuesta en la etapa anterior es la adecuada, es decir, su validez debe ser probada con la realidad, debe ser aprobada con el usuario final.

Es por ello que se plantea realizar las siguientes tareas en esta etapa:

1. **Evaluar**

Acá se evalúa o revisa el valor de la solución del requisito obtenido en la etapa *Reutilizar* (el contenido del documento SR). En este caso por ejemplo se va evaluar si el valor del atributo que tiene la solución, por ejemplo: SR (Especificación del Requisito), es una solución para nuestro problema *r'*.

Luego se debe validar si la solución del requisito es conforme a la realidad del problema, es decir, si es conforme a las exigencias del usuario. Esta tarea se realiza para saber si este requisito cumple con las exigencias de los usuarios.

2. **Reparar**

Si hay algunos cambios que se tiene que realizar a la solución obtenida, es decir, al documento SR, se puede modificar dicha solución, se modifica este documento de acuerdo a las necesidades del usuario, luego se pasará nuevamente por la tarea *Evaluar* (ver Anexos 6 y 7).

3. **Descartar**

Si después de reparar o modificar la solución aún no está conforme la solución con la realidad o si se tiene que realizar demasiados cambios al documento SR, se descarta el requisito.

A continuación se plantea el algoritmo que realiza todas las tareas expuestas anteriormente (ver Tabla 4.5):

Tabla 4.5. *Algoritmo para revisar requisitos.*

Revisar _ Requisito

Inicio

// Entrada: *ri* = {f *i* 1, f *i* 2, f *i* 3, f *i* 4, f *i* 5,.. f*i* 10} Requisito resuelto

// Salida: *ri* revisado o, 0: Requisito descartado

1. ***Evaluar*** la el valor de solución de requisito *ri:*
   SI *fi 10* está conforme a la realidad
   Entonces *ri* será la solución *de r'* // ir a la siguiente etapa: Retener
   SINO
2. ***Reparar*** *fi 10*
   SI *fi 10 reparado es conforme*
3. Entonces *Ir al paso 1*
   SINO
4. **Descartar** Requisito //Salir

Fin

- ***Retener***

Tal como se muestra en la Figura 4.1, luego de tener el requisito revisado se plantea realizar las siguientes tareas:

- En la tarea Extraer atributos del requisito, se analiza qué información que proporciona el requisito revisado *ri* se usará para nuestro problema *r'*, es decir, qué atributos del requisito *ri* tomaremos como parte de nuestra solución para el requisito *r'*, para luego incorporar dicha información a *r'*. En nuestro caso sólo tomaremos el atributo *SR*.

- En la tarea Indexar nuevo requisito, se debe indexar el nuevo requisito *r'* para que el futuro se pueda acceder a él sin problemas. Para esta tarea se usará el algoritmo que se muestra en la Tabla 4.2.

- Luego se verifica nuevamente si la información del nuevo requisito *r'* (ya incorporado con la solución de *ri*) es conforme con la realidad. El analista tiene que revisar nuevamente el documento (SR) con el usuario final.

- Integrar requisito en la LR. Si la solución de *r'* es conforme con la realidad, se integra a la Librería de requisito. Si el requisito *r'* no es conforme, vuelve la etapa *Recuperar*, para seguir el ciclo CBR nuevamente. El analista decide si el nuevo requisito se integrará a la LR.

A continuación se plantea el algoritmo que realiza todas las tareas expuestas anteriormente (ver Tabla 4.6):

Tabla 4.6. Algoritmo para retener requisitos.

Retener _ Requisito

Inicio

// Entrada: *ri* = {f *i* 1, f *i* 2, f *i* 3, f *i* 4, f *i* 5,.. f*i* 10} Requisito revisado

// Salida: *r'* retenido o para la siguiente etapa.

1. Extraer atributos del requisitos *ri*
2. Copiar estos atributos al requisito *r'*
3. Indexar nuevo requisito *r' //* usar algoritmo de indexación
4. Revisar requisito *r'*
5. **Si** *r'* está conforme
   Entonces incorporarlo a la LC
   Sino
6. Ir a la etapa *Recuperar*

Fin

CAPÍTULO 5

UNA HERRAMIENTA PARA LA CAPTACIÓN DE REQUISITOS APLICANDO CBR

El presente capítulo describe el inicio del aporte práctico, donde se presenta todos los pasos a seguir para la construcción y el uso de una herramienta que permitirá la implementación del modelo propuesto en el capítulo anterior. Por ello, esta sección se compone de dos partes: en la primera parte se describe cómo se ha implementado la herramienta, y en la segunda parte se describe el uso de la herramienta.

5.1 Desarrollo de la Herramienta

Esta herramienta se ha desarrollado tomando como base el ciclo de vida CBR de [Aamodt+ 1994]. Se ha utilizado el software base Oracle Form Builder 6.0 y para la Librería de Casos se ha utilizado la base de datos Oracle 10g.

A continuación se presenta algunas de las actividades relacionadas con el análisis y diseño en la implementación de la herramienta.

5.1.1 Construcción del Diagrama de Paquetes

La Figura 5.1 muestra los paquetes con los que contará la herramienta. El paquete Diseño *CBR* contiene los casos de usos relacionados con todas las tareas en la construcción de CBR, el paquete *Ciclo CBR* contiene los casos de usos relacionados con todas las tareas del ciclo CBR, el paquete *Ing. Requisitos* contiene los casos de usos relacionados con la captación de requisitos y el paquete

Consultas contiene los casos de usos relacionados con todas las consultas de los requisitos que se encuentran en nuestra librería de casos.

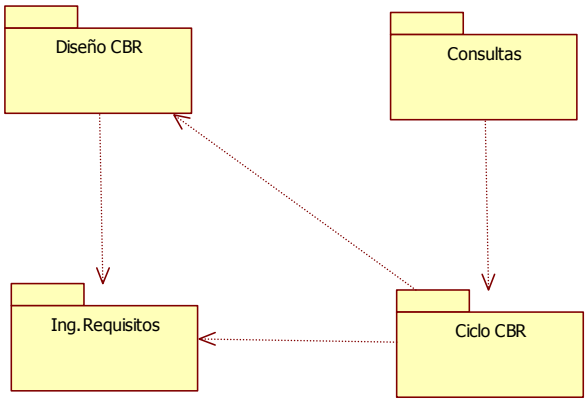


Figura 5.1. Paquetes de la herramienta.

5.1.2 Construcción del Diagrama de Casos de Usos

La Figura 5.2 muestra los casos de usos relacionados con el estudio del dominio y la captación de requisitos. Tal como se muestra en la Figura 5.2, el actor (analista) interactúa con los casos de usos: *Estudio del dominio, Capturar requisito con CBR* y *Capturar requisito con otra técnica.*

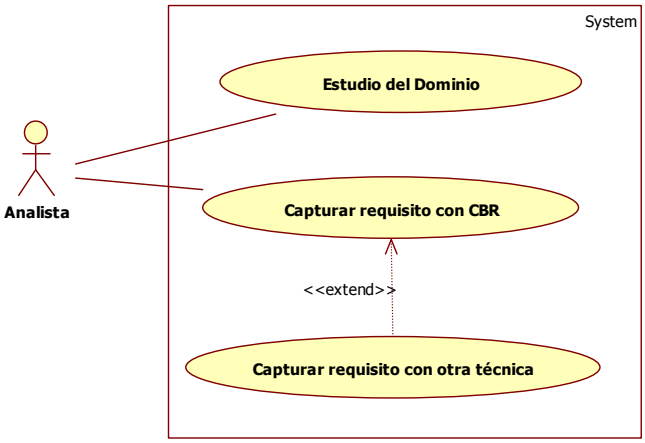


Figura 5.2. Diagrama de Caso de uso del paquete *Ing. Requisitos.*

La Figura 5.3 muestra los casos de usos relacionados con el diseño de CBR. Tal como se muestra en la Figura 5.3, el actor (analista) interactúa con los casos de

usos: *Representar requisito, Definir modelo de memoria, Definir técnica de indización* y *Llenar librería de requisito.*

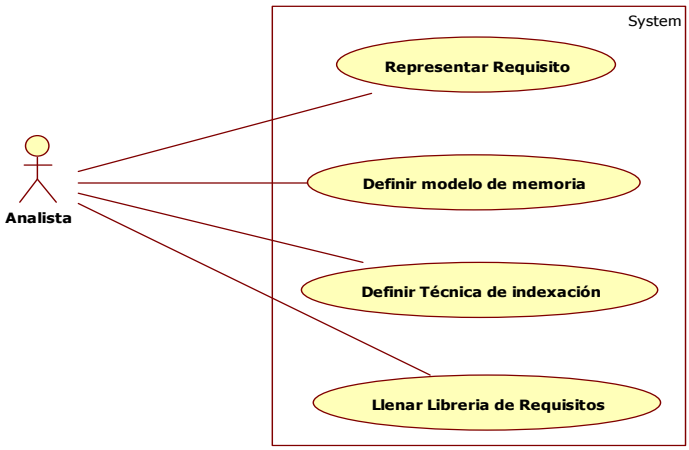


Figura 5.3. Diagrama de Caso de uso del paquete *Diseño CBR.*

La Figura 5.4 muestra los casos de usos relacionados con las tareas del ciclo de vida CBR. Tal como se muestra en la Figura 5.4, el actor (analista) interactúa con los casos de usos: *Identificar requisito similar, Usar técnica de similitud, Modificar solución, Seleccionar requisito similar, Copiar, Revisar* y *Retener requisito.*

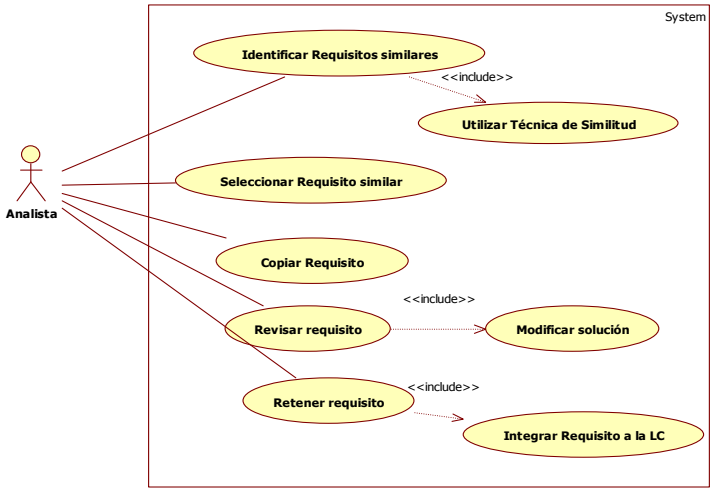


Figura 5.4. Diagrama de Caso de uso del paquete *Ciclo CBR.*

5.1.3 Construcción del Diagrama de clases

La Figura 5.5 muestra el diagrama de clases de la herramienta. En el diagrama se puede observar tres clases: *LibreriaRequisitos*, *Requisitos* y *AtributosRequisito*, cada clase con sus respectivos atributos y operaciones.

La clase *LibreriaRequisitos* representa la librería de casos donde se almacenarán todos nuestros requisitos. La clase *Requisitos* representa los casos o requisitos que vamos a capturar. La clase *AtributosRequisito* representa todos los atributos de los requisitos, por ejemplo: nombre, tipo, pesos, etc. Finalmente, la clase *Proyecto* representa los proyectos o sistemas a los que pertenecen los requisitos.

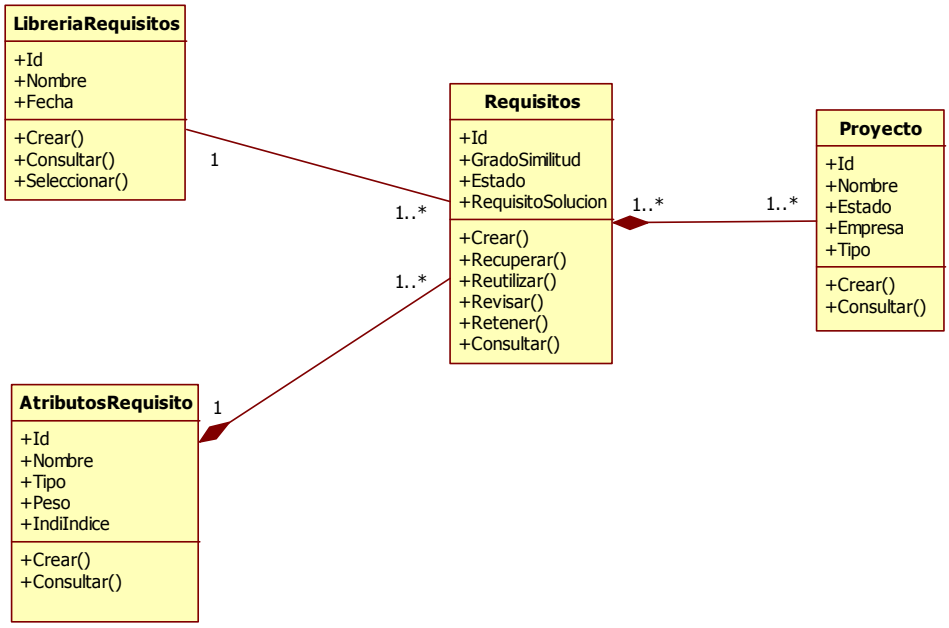


Figura 5.5. Diagrama de Clases de la herramienta.

5.2 Uso de la Herramienta

5.2.1 Registro del Estudio del dominio

Para el análisis del estudio del dominio del negocio se propone usar las notaciones gráficas BPMN para definir los diagramas de flujo de los procesos de negocio, tal como se mencionó en al capítulo anterior. Para ello se usará la herramienta gráfica BizAgi Process Modeler (ver Figura 5.6).

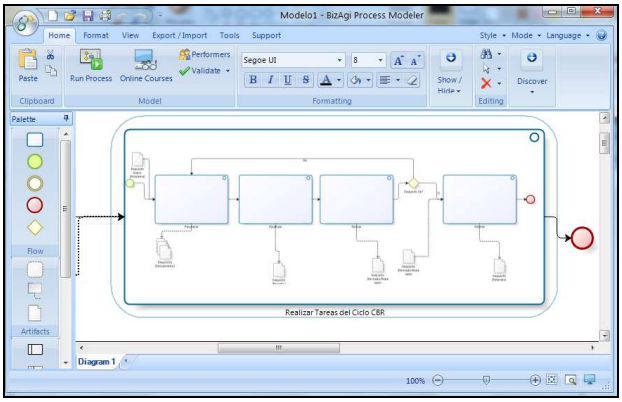


Figura 5.6. BizAgi Process Modeler**.**

5.2.2 Representación del Requisito

Para la representación del caso, es decir, de nuestro requisito, debemos configurar todos sus atributos: etiquetas, pesos y valor, tal como se explicó en la propuesta de nuestro modelo. Para ello seleccionamos la opción Representación del Requisito de la herramienta (ver Figura 5.7).

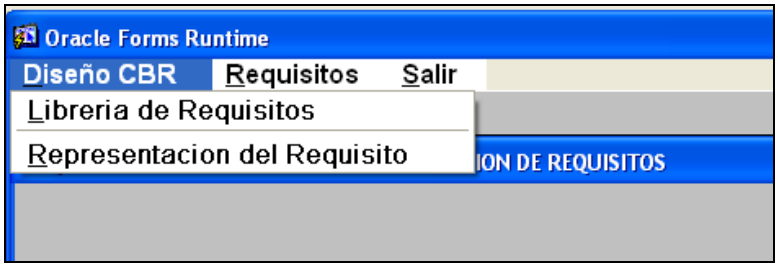


Figura 5.7. Opciones de la herramienta.

Un ejemplo se muestra en la Figura 5.8, donde se ingresan todos los atributos de nuestro caso. Para una búsqueda eficiente se debe ingresar los pesos de cada uno de los atributos, ya que esta herramienta usa la fórmula de Distancia Euclidean Ponderada para calcular la similitud de cada uno de los requisitos, tal como se propone en la sección anterior del presente trabajo.

Figura. 5.8 Representación del Requisito.

5.2.3 Creación de la Librería de Requisitos

Para poder diseñar nuestro modelo lo primero que se debe hacer es crear la Librería de Casos. Para ello se debe escoger la opción Librería de Casos de la herramienta.

A continuación aparecerá la siguiente ventana (ver Figura 5.9). Si todavía no se ha creado la LR se deberá escoger la opción “Agregar”, colocar el nombre de la Librería de Requisitos y guardar los cambios presionando el botón “Guardar”. Si ya se creó la Librería de Requisitos, sólo deberá seleccionar la Librería de Requisitos marcando la opción “Escoger “.

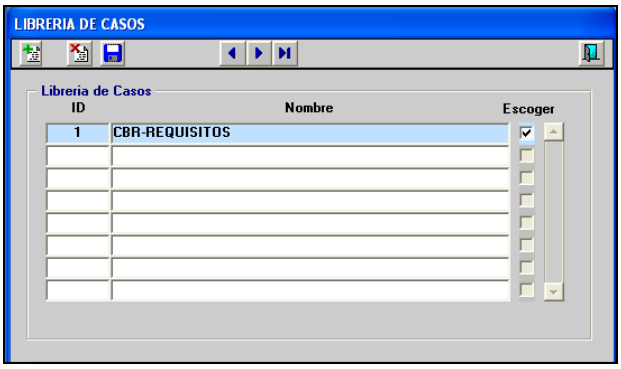


Figura. 5.9 Creación de la Librería de Requisitos.

5.2.4 Llenar la Librería de Requisitos

Para llenar la *Librería de Requisitos* con casos o requisitos de otros proyectos, se correrá un script de base de datos.

5.2.5 Recuperar y Reutilizar el Requisito

Para recuperar y luego reutilizar los requisitos, se debe entrar a la opción *Recuperar-Reutilizar* de la herramienta. Como se explicó en el punto anterior, para recuperar los requisitos similares que se encuentran en nuestra LC, primero se debe ingresar el requisito o caso que se desea resolver (ver Figura 5.10).

Figura 5.10. Opciones de la herramienta.

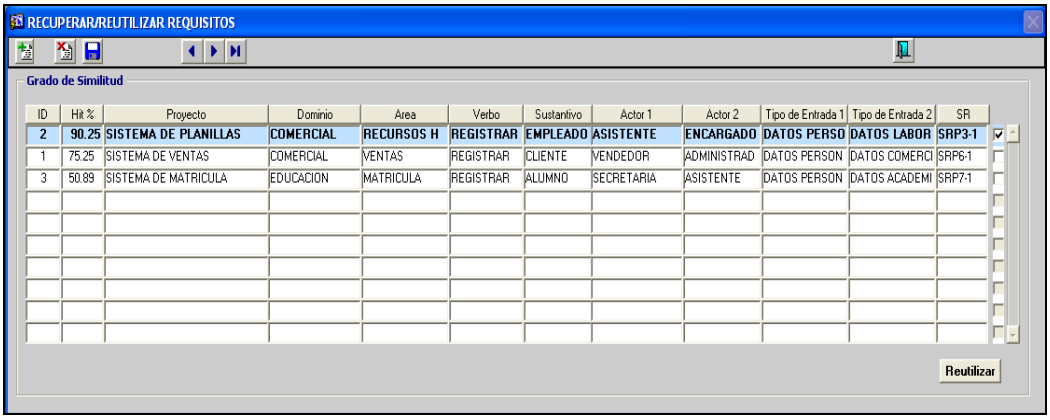
Un ejemplo se muestra en la Figura 5.11, donde se quiere hallar la solución del requisito cuyos atributos son: Nombre del Proyecto (Sistema Integrado de Recursos Humanos), Dominio (Comercial), Área (Recursos Humanos), Verbo (Registrar), Sustantivo (Persona), Actor1 (Asistente), Actor 2 (Jefe), Tipo de Entrada 1 (Datos personales), tipo de entrada 2 (Datos laborales).

Tal como explicamos en la propuesta de nuestro modelo, la solución del requisito estará representada por el atributo SR (Especificación del Requisito), es por ello que este atributo se denota con el carácter "?" (en blanco).

Luego de ingresar los atributos del requisito o problema que se desea resolver, se debe ingresar el número de casos similares que se quiere mostrar. Para ello ingresamos este número en el campo: Máximo Nro. de casos a recuperar, y seguidamente presionamos el botón *Buscar Requisitos* (ver Figura 5.11).

Figura 5.11. Recuperar Requisito.

Después del paso anterior, la herramienta mostrará la ventana dada en la Figura 5.12, con los resultados de la búsqueda, en orden a la similitud de cada uno de los requisitos almacenados en la librería de casos.



ID	Hit %	Proyecto	Dominio	Area	Verbo	Sustantivo	Actor 1	Actor 2	Tipo de Entrada 1	Tipo de Entrada 2	SR
2	90.25	SISTEMA DE PLANILLAS	COMERCIAL	RECURSOS H	REGISTRAR	EMPLEADO	ASISTENTE	ENCARGADO	DATOS PERSO	DATOS LABOR	SRP3-1
1	75.25	SISTEMA DE VENTAS	COMERCIAL	VENTAS	REGISTRAR	CLIENTE	VENDEDOR	ADMINISTRAD	DATOS PERSON	DATOS COMERCI	SRP6-1
3	50.89	SISTEMA DE MATRICULA	EDUCACION	MATRICULA	REGISTRAR	ALUMNO	SECRETARIA	ASISTENTE	DATOS PERSON	DATOS ACADEMI	SRP7-1

Figura. 5.12 Resultados de Recuperar y reutilizar el Requisito.

Para reutilizar la información del caso o requisito que es similar o igual a nuestro problema, se debe seleccionar el requisito que tenga un buen porcentaje de similitud (Hit %). Para ello se debe marcar el requisito y presionar el botón Reutilizar, un ejemplo de esto se muestra en la Figura 5.12. Luego de presionar el botón Reutilizar, el sistema preguntará si está seguro de reutilizar este caso. Finalmente, la información de este caso se utilizará como parte de la solución de nuestro requisito.

5.2.6 Revisar el Requisito

Para que la posible solución del requisito sea revisado por un analista se debe cambiar el estado de Reutilizado a Revisado. Para ello se ingresará a la opción Revisar de la herramienta.

Un ejemplo se muestra en la Figura 5.13, en cuya ventana se observa que el caso que queremos resolver se encuentra en el estado Reutilizado y está asignado el ID del caso que estamos usando como posible solución. Para cambiar el estado del requisito Revisado sólo se debe escoger la opción Estado y seleccionar el estado *Revisado*.

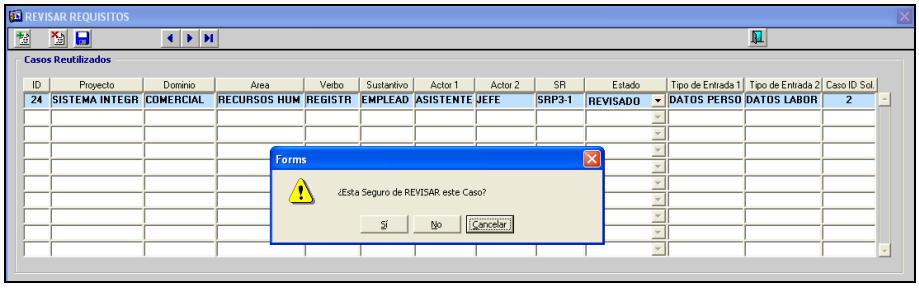


Figura. 5.13. Revisión del Requisito.

Como ahora el requisito se encuentra en estado Revisado, el analista podrá verificar la información del SR del caso escogido como posible solución. Si esta información es la correcta, este atributo se usará como posible solución sin ningún cambio. En caso contrario, si se tiene que hacer algunas modificaciones al SR, este atributo se guardará como parte de la solución del requisito pero con otro nombre. Luego el requisito podrá pasar a la siguiente etapa.

5.2.7 Retener el Requisito en la Librería de Casos

Finalmente, si el requisito ya fue revisado por el analista podrá pasar a esta etapa y se debe cambiar el estado de Revisado a Retenido. Para ello se ingresará a la opción Revisar de la herramienta.

Un ejemplo se muestra en la Figura 5.14, en esta ventana se observa que el caso que queremos resolver se encuentra en el estado Revisado. Para cambiar el estado del requisito Retenido sólo se debe escoger la opción Estado y seleccionar el estado Retenido. De este modo, la información del requisito se copiará en la LC con la solución obtenida.

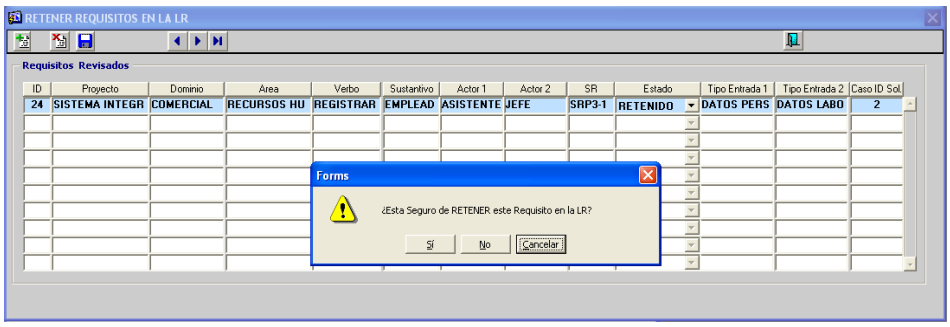
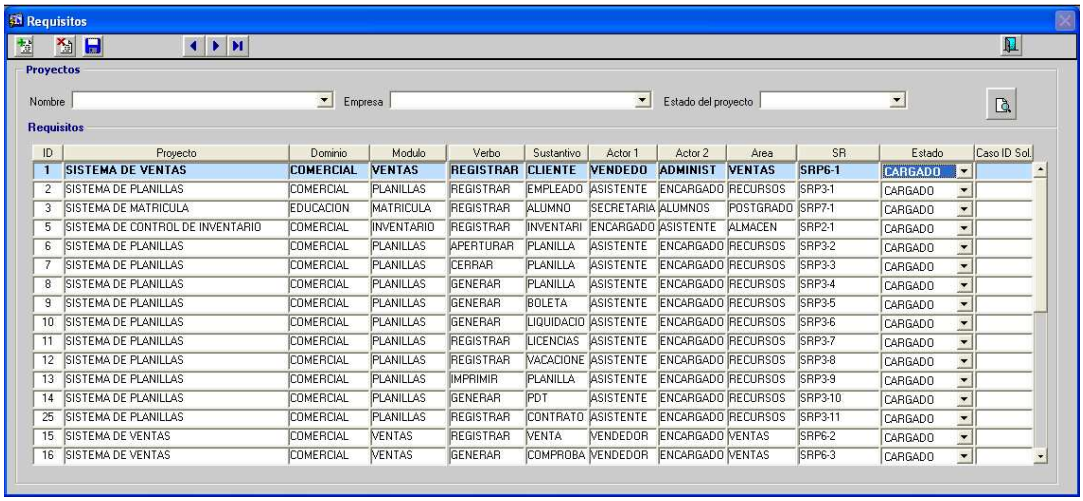


Figura. 5.14. Retención del Requisito.

5.2.8 Consultar Requisitos en la Librería de Casos

Si escogemos la opción *Casos* de la herramienta, podemos consultar todos los casos de nuestra LC. Por ejemplo, podemos observar los requisitos o casos que han sido cargados a la Librería de Requisitos. Estos requisitos se pueden filtrar por: Proyecto, Empresa y Estado del proyecto (ver Figura 5.15).



ID	Proyecto	Dominio	Modulo	Verbo	Sustantivo	Actor 1	Actor 2	Area	SR	Estado	Caso ID Sol.
1	SISTEMA DE VENTAS	COMERCIAL	VENTAS	REGISTRAR	CLIENTE	VENDEDO	ADMINIST	VENTAS	SRP6-1	CARGADO	
2	SISTEMA DE PLANILLAS	COMERCIAL	PLANILLAS	REGISTRAR	EMPLEADO	ASISTENTE	ENCARGADO	RECURSOS	SRP3-1	CARGADO	
3	SISTEMA DE MATRICULA	EDUCACION	MATRICULA	REGISTRAR	ALUMNO	SECRETARIA	ALUMNOS	POSTGRADO	SRP7-1	CARGADO	
5	SISTEMA DE CONTROL DE INVENTARIO	COMERCIAL	INVENTARIO	REGISTRAR	INVENTARI	ENCARGADO	ASISTENTE	ALMACEN	SRP2-1	CARGADO	
6	SISTEMA DE PLANILLAS	COMERCIAL	PLANILLAS	APERTURAR	PLANILLA	ASISTENTE	ENCARGADO	RECURSOS	SRP3-2	CARGADO	
7	SISTEMA DE PLANILLAS	COMERCIAL	PLANILLAS	CERRAR	PLANILLA	ASISTENTE	ENCARGADO	RECURSOS	SRP3-3	CARGADO	
8	SISTEMA DE PLANILLAS	COMERCIAL	PLANILLAS	GENERAR	PLANILLA	ASISTENTE	ENCARGADO	RECURSOS	SRP3-4	CARGADO	
9	SISTEMA DE PLANILLAS	COMERCIAL	PLANILLAS	GENERAR	BOLETA	ASISTENTE	ENCARGADO	RECURSOS	SRP3-5	CARGADO	
10	SISTEMA DE PLANILLAS	COMERCIAL	PLANILLAS	GENERAR	LIQUIDACIO	ASISTENTE	ENCARGADO	RECURSOS	SRP3-6	CARGADO	
11	SISTEMA DE PLANILLAS	COMERCIAL	PLANILLAS	REGISTRAR	LICENCIAS	ASISTENTE	ENCARGADO	RECURSOS	SRP3-7	CARGADO	
12	SISTEMA DE PLANILLAS	COMERCIAL	PLANILLAS	REGISTRAR	VACACIONE	ASISTENTE	ENCARGADO	RECURSOS	SRP3-8	CARGADO	
13	SISTEMA DE PLANILLAS	COMERCIAL	PLANILLAS	IMPRIMIR	PLANILLA	ASISTENTE	ENCARGADO	RECURSOS	SRP3-9	CARGADO	
14	SISTEMA DE PLANILLAS	COMERCIAL	PLANILLAS	GENERAR	PDT	ASISTENTE	ENCARGADO	RECURSOS	SRP3-10	CARGADO	
25	SISTEMA DE PLANILLAS	COMERCIAL	PLANILLAS	REGISTRAR	CONTRATO	ASISTENTE	ENCARGADO	RECURSOS	SRP3-11	CARGADO	
15	SISTEMA DE VENTAS	COMERCIAL	VENTAS	REGISTRAR	VENTA	VENDEDOR	ENCARGADO	VENTAS	SRP6-2	CARGADO	
16	SISTEMA DE VENTAS	COMERCIAL	VENTAS	GENERAR	COMPROBA	VENDEDOR	ENCARGADO	VENTAS	SRP6-3	CARGADO	

Figura. 5.15. Casos de la LR

CAPÍTULO 6

CASOS DE ESTUDIO

En el presente capítulo se analizarán dos casos de estudios, donde se seguirá cada uno de los pasos expuestos en el capítulo 4. Primero se aplicará a cada caso de estudio: *Casos de Usos, Win Win* y el *modelo CBR propuesto* para captar los requisitos. Luego se aplicará métricas de calidad a los requisitos obtenidos con cada una de las técnicas y el *modelo CBR propuesto*, para finalmente comprobar los resultados.

6.1 Contexto general de los casos de estudio

6.1.1 Caso de estudio 1: Sistema Integrado de Recursos Humanos (SIRH)

El primer caso de estudio trata sobre el *Sistema Integrado de Recursos Humanos (SIRH)*, que se desarrolló para la Caja de Pensiones Militar Policial en el año 2004.

Este Sistema es una solución administrativa y funcional que procesa la información de la empresa, aportando productividad, flexibilidad y descentralización a los controles financieros relacionados con el manejo del personal.

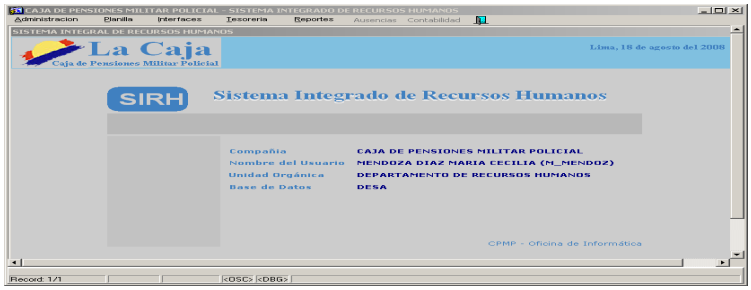


Figura. 6.1. Interface gráfica del Sistema Integrado de Recursos Humanos (SIRH).

En la Tabla 6.1 se muestra todas las funcionalidades de cada uno de los módulos del *SIRH*.

Tabla 6.1. Módulos del Sistema Integrado de Recursos Humanos (SIRH).

Módulos				
Administración	**Planillas**	**Interfaces**	**Tesorería**	**Reportes**
▪ Selección de puestos de trabajos. ▪ Administración de Personal, registro de información de Empleados (datos personales, laborales, remuneraciones actuales e históricas, datos familiares, datos de estudios, etc.). ▪ Administración de Contratos: múltiples tipos, emisión, control de vencimientos y renovaciones, tanto de Empleados, Locadores y practicantes. ▪ Administración Capacitación: registro detallado de cursos, empresas, costos, capacitación por área y trabajador. ▪ Evaluación de desempeño: evaluación del desempeño de los empleados y funcionarios. ▪ Control de Sanciones y Amonestaciones.	▪ Maestros de códigos del sistema, rubros, conceptos/asientos y parámetros de la planilla. ▪ Registro de datos no calculados: ingresos, descuentos. ▪ Procesos: generación de adelantos de Planilla, adelantos de Vacaciones primer y segundo periodo, generación de Compensación Vacacional, generación de Planilla, Cierre, Liquidaciones. ▪ Administración de Vacaciones y Licencias. ▪ Descuentos al personal.	▪ Información que generan entradas al cálculo de planilla: descuentos por vales de cafetería, ausencias, tardanzas, etc. ▪ Información que se genera después del cálculo de planilla: CTS, AFP, Pagos Judiciales, archivos de abono, generación de provisiones a contabilidad.	▪ Generación de Solicitudes de pago de honorarios y pagos diversos (dietas al directorio, Rendición de Cuentas, etc.), eliminación de Solicitudes de pago.	▪ Pantallas organizadas de reportes de empleados, planillas, CTS, Quinta, AFP, Ley, Vacaciones, Liquidaciones, Judiciales, cafetería, ausencias, capacitación, etc.

6.1.2 Caso de estudio 2: Sistema de Punto de Venta (SPTA)

El segundo caso de estudio trata sobre el *Sistema de Punto de Venta (SPTA)*, que se desarrolló para la cadena de farmacias Boticas FASA en el año 2003.

Este Sistema es una solución que permite controlar las ventas de cada local, las ventas hechas por Delivery, el Inventario y Stock de productos, los Traspasos de mercadería que se realizan entre locales o puntos de ventas, etc.

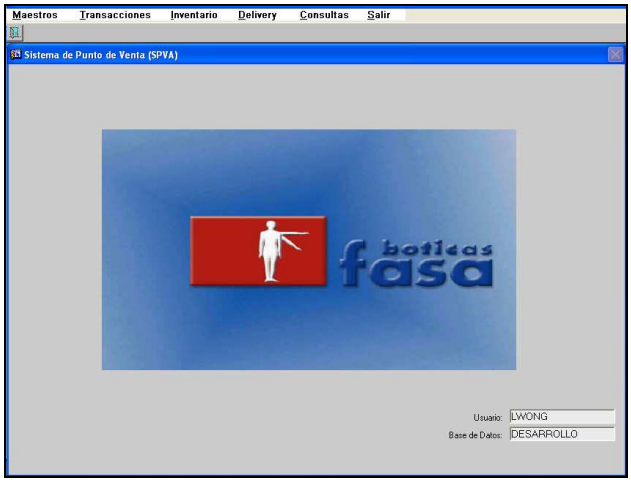


Figura. 6.2. Interface gráfica del Sistema de Punto de Venta (SPTA).

En la Tabla 6.2 se muestra todas las funcionalidades de cada uno de los módulos del SIRH.

Tabla 6.2. Módulos del Sistema Integrado de Recursos Humanos (SIRH).

Módulos				
Maestros	**Transacciones**	**Consultas**	**Inventarios**	**Delivery**
▪ Productos ▪ Acciones terapéuticas ▪ Posología de los medicamentos ▪ Serie de Comprobantes ▪ Maestro de Motorizados ▪ Motivos de Inventario ▪ Motivos de Reprogramación ▪ Motivo de Regularización	▪ Traspasos de mercadería ▪ Devoluciones de mercaderías por locales ▪ Recepción de guías de almacén ▪ Corrección de guías ▪ Sugeridos de precios máximos y mínimos de los productos ▪ Pedidos al centro de despacho ▪ Balancín diario de las ventas por locales ▪ Envío de Stock ▪ Envío de Traspasos ▪ Corrección de documentos	▪ Traspasos ▪ Devoluciones ▪ Ventas de Locales ▪ Ventas Delivery ▪ Recepción de guías de almacén ▪ Productos por guías transferidas ▪ Stock de productos ▪ Stock de cadena ▪ Kárdex de Productos ▪ Máximos y Mínimos	▪ Inicio de Inventario ▪ Carga de Pistolas ▪ Descarga de Pistolas	▪ Recepción de Pedidos ▪ Tipos de Pedidos ▪ Registro de Impresión de Tickets

6.2 Métricas de Calidad

A continuación se presenta las métricas de calidad que se van a aplicar a los resultados obtenidos de los casos de estudio y, de esta manera, se observará la calidad de requisitos identificado con nuestro modelo.

6.2.1 *Métrica de Calidad: Claridad de Descripción*

La presente métrica se refiere a la característica *Usabilidad* [ISO 2011], donde se medirá el grado de *Entendibilidad* de los requisitos, es decir, se verificará si los requisitos obtenidos con el modelo propuesto son entendibles por el usuario (ver Tabla 6.3).

Tabla 6.3. Métrica de calidad aplicada a la Usabilidad de los requisitos identificados con el modelo CBR propuesto [ISO 2011].

Nombre de la Métrica:	Claridad de la descripción
Código:	MC001
Característica:	Usabilidad
Subcaracterística:	Entendibilidad
Propósito:	¿Qué proporción de los requisitos es entendida después de leer las especificaciones?
Método de aplicación:	Contar el número de los requisitos que se entienden adecuadamente y comparar con el número total de requisitos.
Medición, fórmula y elementos medibles:	X = A / B
	A = Nro. de requisitos entendidos
	B = Total de Nro. de requisitos
Tipo de medida:	A = Contador
	B = Contador
Interpretación del valor medido:	0 < = X < = 1
	Lo más cercano a 1 es lo mejor
Tipo de escala:	Absoluta

6.2.2 *Métrica de Calidad: Completitud Funcional*

La presente métrica se refiere a la característica *Funcionalidad* [ISO/IEC 25010], donde se medirá el grado de *Completitud Funcional* de los requisitos, es decir, se verificará si los requisitos obtenidos con el modelo propuesto satisfacen a las necesidades de los usuarios (ver Tabla 6.4).

Tabla 6.4. Métrica de calidad aplicada a la Usabilidad de los requisitos identificados con el modelo CBR propuesto [ISO 2011].

Nombre de la Métrica:	Conformidad con la Completitud Funcional
Código:	MC002
Característica:	Funcionalidad
Subcaracterística:	Completitud Funcional
Propósito:	¿Qué proporción de los requisitos satisface en un 100 % las necesidades del usuario?
Método de aplicación:	Contar el número de requisitos que satisfagan las necesidades del usuario en un 100% y comparar con el número total de requisitos.
Medición, fórmula y elementos medibles:	X = A / B
	A = Nro. de requisitos que satisfacen en un 100% las necesidades del usuario
	B = Total de Nro. de requisitos
Tipo de medida:	A = Contador
	B = Contador
Interpretación del valor medido:	0 < = X < = 1
	Lo más cercano a 1 es lo mejor
Tipo de escala:	Absoluta

6.2.3 *Métrica de Calidad: Reusabilidad*

La presente métrica se refiere a la característica *Mantenibilidad* [ISO/IEC 25010], donde se medirá el grado de *Reusabilidad* de los requisitos, es decir, se obtendrá el grado de reusabilidad de los requisitos obtenidos con el modelo propuesto (ver Tabla 6.5).

Tabla 6.5. Métrica de calidad aplicada a la Mantenibilidad de los requisitos identificados con el modelo CBR propuesto [ISO 2011].

Nombre de la Métrica:	Conformidad de Reusabilidad
Código:	MC003
Característica:	Mantenibilidad
Subcaracterística:	Reusabilidad
Propósito:	¿Qué cantidad de requisitos son reusados para dar solución a nuevos problemas?
Método de aplicación:	Contar el número de requisitos que son reusados para identificar otros requisitos y comparar con el número total de requisitos.
Medición, fórmula y elementos medibles:	X = A / B
	A = Nro. de requisitos reusados
	B = Total de Nro. de requisitos
Tipo de medida:	A = Contador
	B = Contador
Interpretación del valor medido:	0 < = X < = 1
	Lo más cercano a 1 es lo mejor
Tipo de escala:	Absoluta

6.2.4 Métrica de Calidad: Eficiencia

La presente métrica se refiere a la característica *Eficiencia* [ISO/IEC 25010], donde se medirá el *Tiempo* en minutos que se demora en identificar el requisito, es decir, se obtendrá el tiempo promedio en la identificación de los requisitos utilizando el modelo CBR propuesto (ver Tabla 6.6).

Tabla 6.6. Métrica de calidad aplicada a la eficiencia de los requisitos identificados con el modelo CBR propuesto [ISO 2011].

Nombre de la Métrica:	Tiempo de demora
Código:	MC004
Característica:	Eficiencia
Subcaracteristica:	Tiempo
Propósito:	¿Cuál es el tiempo promedio que demora en identificar los requisitos utilizando el modelo propuesto?
Método de aplicación:	Calcular el tiempo promedio en la identificación de los requisitos utilizando el modelo CBR propuesto y compararlos con otras técnicas.
Medición, fórmula y elementos medibles:	$X = \sum Ti / B$
	T = Tiempo que demora en identificar el requisito usando el modelo CBR propuesto
	B = Total de Nro. de requisitos
Tipo de medida:	T = Tiempo en minuto
	B = Contador

6.3 Pruebas del caso de estudio 1

En el presente caso de estudio se escogieron diez requisitos funcionales del *Sistema de* Recursos *Humanos (SIRH)* para realizar la captación de requisitos, donde se aplicaron: *Casos de Uso*, *Win Win* y el modelo *CBR* propuesto.

En la Tabla 6.7 se muestra los diez requisitos funcionales del *SIRH* que se han escogido. Como se puede observar, se seleccionaron los requisitos que tienen prioridad *Alta*. La prioridad de los requisitos fue realizada por el usuario final.

Tabla 6.7. Requisitos funcionales prioritarios del Sistema Integrado de Recursos Humanos (SIRH).

Código	Requisitos	Prioridad
R1	Registrar Empleado	Alta
R2	Generar Planilla	Alta
R3	Generar Boleta	Alta
R4	Aperturar Planilla	Alta
R5	Cerrar Planilla	Alta
R6	Generar Liquidaciones	Alta
R7	Registrar Licencias	Alta
R8	Registrar Vacaciones	Alta
R9	Imprimir Planilla	Alta
R10	Generar PDT	Alta

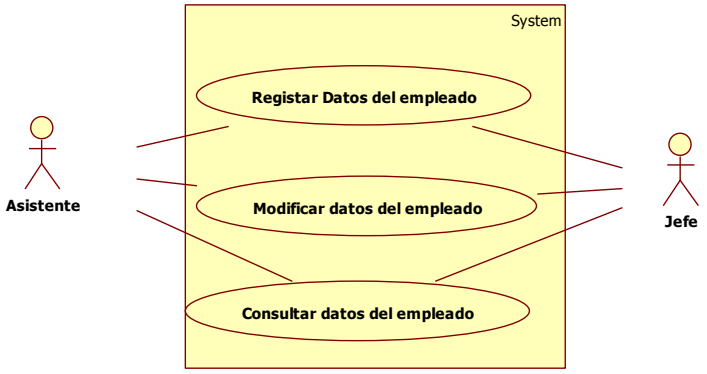
6.3.1 Captación de Requisitos aplicando Casos de Uso

En esta sección se va capturar los requisitos de este caso de estudio, aplicando la técnica *Casos de Usos* explicados en el capítulo 3. La Tabla 6.8 muestra un resumen del *Caso de Uso* del requisito*: R1-Registrar Empleado.* Allí se detalla el diagrama de Casos de Uso, una breve descripción del caso de uso y los actores que intervienen.

Tabla 6.8. Caso de Uso del requisito *R1-Registrar Empleado* del caso de estudio 1.

Caso de Uso: R1- Registrar empleado	**Analista:** Ediberto Iparraguirre

Diagrama de Casos de Uso



Breve Descripción

- Este Caso de Uso se refiere a las actividades llevadas a cabo para completar satisfactoriamente el registro de los datos de los empleados, el cual puede ser realizado por el Jefe del Recursos Humanos o por el Asistente de Recursos Humanos.
- El actor podrá registrar los datos del empleado, modificar y consultar su información.

Actores

- Jefe de Recursos Humanos: usuario interno a la organización que está encargado de velar por el cumplimiento de las actividades realizadas con el personal que trabaja en la empresa.
- Asistente de Recursos Humanos: usuario interno a la organización que está encargado de ayudar en las labores y actividades relacionadas con el personal que trabaja en la empresa.

Flujo básico

- El sistema muestra la pantalla para acceder al Sistema.
- El actor ingresa los datos de autenticación para ingresar al sistema.
- El sistema inicia la sesión del actor y le presenta la pantalla de Registro de datos con la pestaña “General” habilitado (Pantalla Registro de datos Generales).
- El actor procede con el llenado de los campos solicitados por el Sistema, los mismos que deberá llenar antes de proceder con los demás pasos. La información que deberá

incluir en este rubro consiste en: Documento de Identidad, Número de Documento, Apellido Paterno, Apellido Materno, Nombres, Nacionalidad, Estado Civil, Sexo (Género), Teléfono, Fax, Email y Dirección.

– El actor presiona el botón "Guardar" para guardar los datos ingresados.

Flujo Alternativo

– El actor decide salir del sistema sin registrar sus datos:
– El actor no ingresa ningún dato y sale de la página donde se encuentra.
– El caso de uso termina.

En la especificación del requisito (SR) se encuentran todos los casos de usos desarrollados. El resumen de los Casos de Usos de los demás requisitos se muestra en el Anexo 2.

6.3.2 Captación de Requisitos aplicando Win Win

En esta sección se va capturar los requisitos de este caso de estudio, aplicando la técnica *Win Win,* explicados en el capítulo 3. Se ha realizado la captación de requisitos en tres iteraciones.

- ***Iteración 1***

En esta fase se tuvo una primera reunión de trabajo, con el Jefe del proyecto, los analistas y stackeholders. En esta reunión se identificaron los requisitos que se muestra en la Tabla 6.9.

Tabla 6.9. Requisitos identificados en la iteración 1 utilizando la técnica Win Win aplicado en el caso de estudio 1.

No.	Requisito	Solicitado por	Cargo	Área	Analista responsable	Estado
1	Registrar Empleado	Cchavez	Asistente	R.R.H.H.	Eiparraguirre	Propuesto
2	Registrar Contratos	Cchavez	Asistente	R.R.H.H.	Eiparraguirre	Propuesto
3	Registrar préstamos	Cchavez	Asistente	R.R.H.H.	Eiparraguirre	Propuesto
4	Aperturar planilla	Cchavez	Asistente	R.R.H.H.	Eiparraguirre	Propuesto

5	Generar Planilla	Cchavez	Asistente	R.R.H.H.	Eiparraguirre	Propuesto
6	Genera Boletas	Cchavez	Asistente	R.R.H.H.	Eiparraguirre	Propuesto
7	Cerrar Planilla	Cchavez	Asistente	R.R.H.H.	Eiparraguirre	Propuesto
8	Generar liquidación	Cchavez	Asistente	R.R.H.H.	Eiparraguirre	Propuesto
9	Registrar Licencias	Cchavez	Asistente	R.R.H.H.	Eiparraguirre	Propuesto
10	Registrar Vacaciones	Cchavez	Asistente	R.R.H.H.	Eiparraguirre	Propuesto
11	Generar reporte de Quinta	Cchavez	Asistente	R.R.H.H.	Eiparraguirre	Propuesto
12	Reporte de vacaciones	Cchavez	Asistente	R.R.H.H.	Eiparraguirre	Propuesto
13	Imprimir Libro de Planilla	Ctorres	Jefe	R.R.H.H.	Eiparraguirre	Propuesto
14	Generar PDT de planillas	Ctorres	Jefe	R.R.H.H.	Eiparraguirre	Propuesto
15	Reporte de AFP	Cchavez	Asistente	R.R.H.H.	Eiparraguirre	Propuesto
16	Reporte de CTS	Cchavez	Asistente	R.R.H.H.	Eiparraguirre	Propuesto
17	Reporte de Vacaciones	Ctorres	Jefe	R.R.H.H.	Eiparraguirre	Propuesto
18	Reporte de Licencias	Ctorres	Jefe	R.R.H.H.	Eiparraguirre	Propuesto
19	Generar reporte vacaciones	Ctorres	Jefe	R.R.H.H.	Eiparraguirre	Propuesto
20	Genera reporte de Liquidaciones	Ctorres	Jefe	R.R.H.H.	Eiparraguirre	Propuesto

- ***Requisitos identificados en la Iteración 2***

En esta segunda fase se tuvo una segunda reunión de trabajo, donde se especificaron con más detalle los requisitos. Los stackeholders dieron prioridades a los requisitos y de acuerdo a estas prioridades se aceptaron o rechazaron los requisitos (ver Tabla 6.10).

Tabla 6.10. Requisitos identificados en la iteración 2 utilizando la técnica Win Win aplicada en el caso de estudio 1.

No.	Requisito	Descripción	Estado	Prioridad
1	Registrar Empleado	El Sistema deberá permitir registrar los datos de los empleados que laboran en la empresa, tales como: nombres y apellidos, tipo de documento de identidad, número de documento de identidad, número telefónico, dirección, correo electrónico, etc.	Aceptado	Alta
2	Registrar Contratos	El Sistema deberá permitir registrar los datos de los contratos de los empleados, practicantes y locadores. Algunos campos como fechas deberán ser actualizadas en los datos del empleado.	Rechazado	Baja
3	Registrar préstamos	El sistema deberá permitir realizar operaciones de préstamos a los empleados de acuerdo a las políticas establecidas por la organización. Además, el sistema permitirá generar el cronograma de pagos respecto del préstamo realizado al empleado.	Aceptado	Media
4	Aperturar planilla	Esta funcionalidad deberá permitir abrir el periodo para luego poder realizar todas las tareas relacionadas al cálculo de la planilla, tales como: adelanto de quincena, adelanto de vacaciones, calcular porcentaje de sueldo para el pago de quincena, etc.	Aceptado	Alta
5	Generar Planilla	El sistema deberá permitir generar los diferentes tipos de Planilla de los empleados, tales como: Planilla Mensual, Planilla de Gratificaciones, Planilla de Vacaciones, etc. Para ello el periodo de la planilla debe estar aperturado. Para generar la planilla se debe calcular el total de ingresos, descuentos y aportaciones por cada empleado. También se debe permitir consultar la planilla.	Aceptado	Alta
6	Genera Boletas	El sistema deberá permitir generar las boletas para cada empleado mensualmente	Aceptado	Alta
7	Cerrar Planilla	El sistema deberá permitir realizar el cierre de la planilla, de un determinado periodo. Para ello se debe de ingresar como parámetro el periodo de la planilla y el tipo de planilla.	Aceptado	Alta
8	Generar liquidación	El sistema deberá permitir realizar el cálculo de la liquidación de un trabajador.	Aceptado	Alta
9	Registrar Licencias	El sistema deberá permitir registrar la programación de Licencia por maternidad, Enfermedad y sin goce de haber. Para ello se debe ingresar el tipo de licencia y los periodos (fecha de inicio y fecha de término). El sistema también debe permitir consultar.	Aceptado	Alta

10	Registrar Vacaciones	El sistema deberá permitir realizar el registro de la programación de vacaciones de los empleados y también realizar las consultas, etc.	Aceptado	Alta
11	Generar reporte de Quinta	El sistema deberá permitir generar el reporte según el descuento de quinta categoría.	Aceptado	Media
12	Reporte de vacaciones	El sistema deberá permitir crear el reporte de vacaciones por trabajador.	Aceptado	Media
13	Imprimir Libro de Planilla	El Sistema deberá permitir generar un reporte del Libro de Planilla, este reporte se generará de forma mensual y contendrá todos los tipos de planilla generados en un determinado periodo.	Aceptado	Alta
14	Generar PDT de planillas	El sistema deberá permitir generar un archivo con información de la planilla de un determinado periodo, para subirlo en el PDT. La generación de este archivo se hará en forma mensual después que se cierre el periodo de la planilla.	Aceptado	Alta
15	Reporte de AFP	El sistema deberá permitir generar el reporte de los descuentos por el concepto de las diferentes AFPs, en un periodo determinado. Para ello se tiene que ingresar el periodo de la planilla.	Aceptado	Media
16	Reporte de CTS	El sistema deberá permitir generar el reporte de los montos por el concepto de CTS de los empleados, en un periodo determinado. Para ello se tiene que ingresar el periodo de la planilla.	Aceptado	Media
17	Reporte de Vacaciones	El sistema deberá permitir generar reportes de acuerdo a determinados parámetros como: inicio y fin de periodo, año, mes, área de los periodos vacacionales ejecutados a la fecha de un determinado empleado.	Aceptado	Media
18	Reporte de Licencias	El sistema deberá permitir generar reportes del total de días de vacaciones comprados por la organización y el total de días de vacaciones efectivizadas por el empleado en un periodo anual.	Aceptado	Media
19	Generar reporte vacaciones	El sistema deberá permitir generar el reporte de los periodos vacacionales que no han sido asignados a ningún empleado.	Aceptado	Media
20	Genera reporte de Liquidaciones	El sistema deberá permitir generar la liquidación de beneficios sociales del empleado.	Aceptado	Media

- ***Requisitos identificados en la Iteración 3***

En esta tercera fase se escogieron los requisitos que tienen prioridad *Alta*. Luego se validaron cada uno de ellos y se realizaron los ajustes necesarios (ver Tabla 6.11).

Tabla 6.11. Requisitos identificados en la iteración 3 utilizando la técnica Win Win aplicado en el caso de estudio 1.

No.	Requisito	Solicitado por	Analista responsable	Validado por	Estado	Prioridad	Estado
1	Registrar Empleado	Cchavez	Eiparraguirre	Cchavez	Validado	Alta	Validado
4	Aperturar planilla	Cchavez	Eiparraguirre	Cchavez	Validado	Alta	Validado
5	Generar Planilla	Cchavez	Eiparraguirre	Cchavez	Validado	Alta	Validado
6	Genera Boletas	Cchavez	Eiparraguirre	Cchavez	Validado	Alta	Validado
7	Cerrar Planilla	Cchavez	Eiparraguirre	Cchavez	Validado	Alta	Validado
8	Generar liquidaciones	Cchavez	Eiparraguirre	Cchavez	Validado	Alta	Validado
9	Registro Licencias	Cchavez	Eiparraguirre	Cchavez	Validado	Alta	Validado
10	Registrar Vacaciones	Cchavez	Eiparraguirre	Cchavez	Validado	Alta	Validado
13	Imprimir Libro de Planilla	Ctorres	Eiparraguirre	Cchavez	Validado	Alta	Validado
14	Generar PDT de planillas	Ctorres	Eiparraguirre	Cchavez	Validado	Alta	Validado

6.3.3 Captación de requisitos usando el Modelo CBR propuesto

En esta sección se realiza la captación de requisitos del mismo caso de estudio, pero aplicando el *modelo CBR propuesto*. A continuación se muestra cada una de las tareas que se tiene que realizar para usar nuestro modelo.

6.3.3.1 Diseñar el modelo CBR propuesto

Para poder realizar los casos de estudio, primero se debe configurar nuestro modelo. Las tareas relacionadas con el diseño del modelo CBR propuesto se realizarán sólo una vez.

A continuación se muestra las tareas que se realizan para el diseño del modelo propuesto:

- ***Crear Librería de Requisitos***

  Para crear la Librería de Requisitos se debe entrar a la opción *"Diseño CBR/Librería de Requisitos"* de la herramienta expuesta en el capítulo 5, el nombre que se da a la Librería de Requisitos es *CBR-REQUISITOS* (ver Figura 6.3).

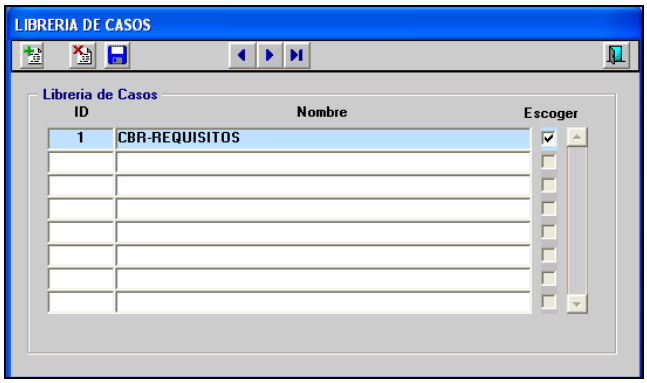


Figura. 6.3. Creación de la Librería de Requisitos.

- ***Llenar Librería de Requisitos***

Para llenar la Librería de Requisitos con requisitos de otros proyectos se ha considerado un script de base de datos con los requisitos funcionales de 8 proyectos. Los proyectos que se han considerado para llenar la Librería de Requisitos se muestra en la Tabla 6.12.

Tabla 6.12. Proyectos considerados en la carga de la *Librería de Requisitos.*

Cód.	Proyecto	Sector	Empresa	Año
P1	Sistema de Pensiones (SPEN)	Comercial	Prima AFP	2010
P2	Sistema de Control de Inventario (SCI)	Comercial	Boticas Fasa	2009
P3	Sistema de Planillas (SPLA)	Comercial	Constructora Queiroz Galvao	2008
P4	Sistema de Préstamos Personales (SPP)	Comercial	Caja de Pensiones Militar Policial	2008
P5	Sistema de Créditos y Cobranzas (SCCO)	Comercial	Caja de Pensiones Militar Policial	2008
P6	Sistema de Ventas (SVTA)	Comercial	Farmacias Goliat	2008
P7	Sistema de Matrícula (SMAT)	Educación	Colegio León Pinelo	2006
P8	Sistema de Información Contable (SIC)	Comercial	Caja de Pensiones Militar Policial	2006

En el Anexo 1 se muestra los proyectos que se han llenado en la *Librería de Requisitos*.

- ***Representar el Requisito***

Para configurar la estructura del requisito, se debe entrar a la opción "*Diseño CBR/Representación de Requisito*" de la herramienta expuesta en el capítulo 5.

Donde se crean nueve atributos, definiendo el nombre, tipo, peso e índice. En la Figura 6.4 se observa que se han creados nueve atributos (Nombre del proyecto, Dominio, Área, Verbo, Sustantivo, Actor 1, Actor 2, Tipo de entrada1, Tipo de entrada 2 y SR), donde el tipo de dato de los atributos es String (caracteres no numéricos).

También se puede observar que el peso del atributo *Dominio* es el que tiene un valor mayor (10), esto quiere decir, que cuando se calcule las distancias entre los atributos, éstos tendrán mayor importancia. Se observa también que los atributos

Dominio y *Área* son índices, esto quiere decir que cuando se realice la búsqueda de los casos se realizará por estos dos atributos.

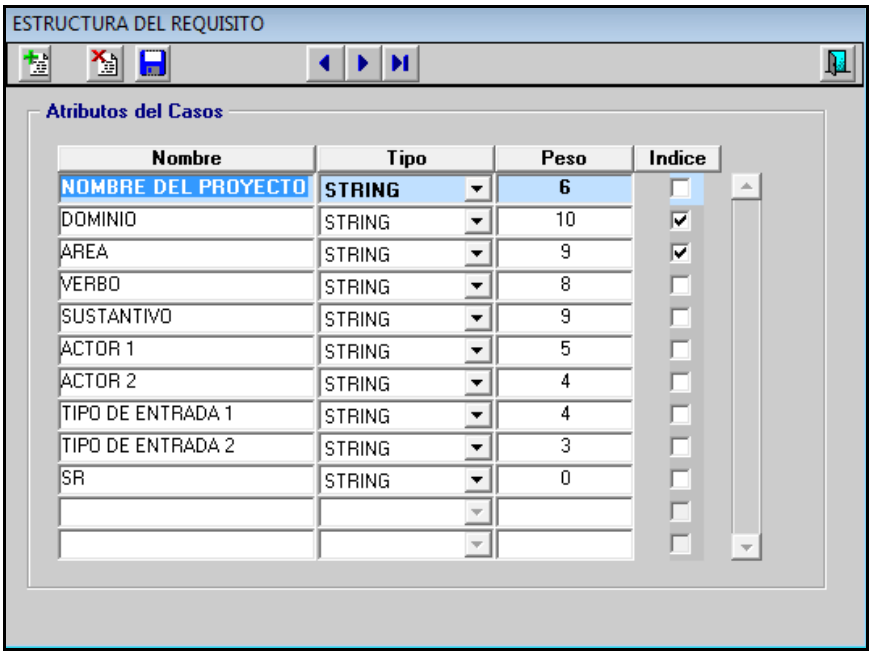


Figura. 6.4. Representación del Requisito.

A continuación se muestran el tiempo en minutos que se invirtió para realizar estas tareas (ver Tabla 6.13).

Tabla 6.13. Tiempo invertido en las tareas para el diseño de CBR.

Tareas para Diseñar CBR	Tiempo (*)
Crear Librería de Requisitos	1
Representar el Requisito	10
Llenar Librería de Requisitos	5
Totales:	16
(*) Tiempo en minutos	

6.3.3.2 Recuperar Requisito

En la Figura 6.5 se muestra cómo se ha recuperado el requisito *R1.Registrar Empleado* del presente caso de estudio. Para ello se tiene que ingresar todos los atributos del nuevo requisito que se quiere resolver. Además, se muestra todos los

atributos, tales como: nombre del proyecto, dominio del proyecto, área, el verbo y el sustantivo del requisito, etc.

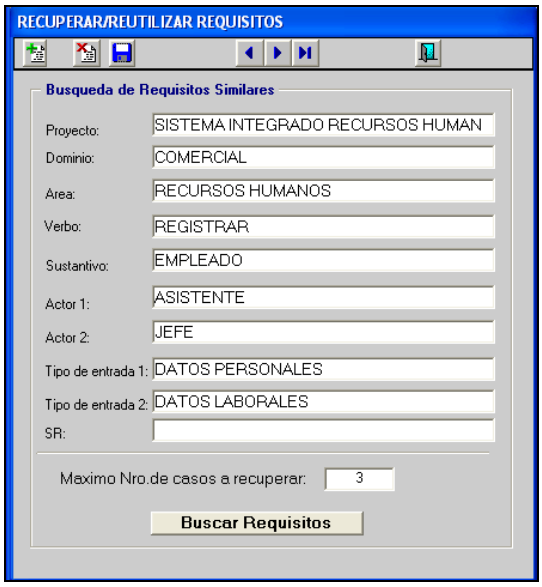


Figura. 6.5. Recuperación del requisito *R1* del caso de estudio 1 utilizando el modelo CBR propuesto.

Después de ingresar los valores de los atributos, se procede a ingresar el número máximo de casos que se quiere recuperar, tal como se muestra en la Figura 6.5, donde se desea que se muestre como máximo tres casos similares a nuestro requisito.

A continuación se muestran los tres requisitos que tienen el mejor grado de similitud (ver Figura 6.6):

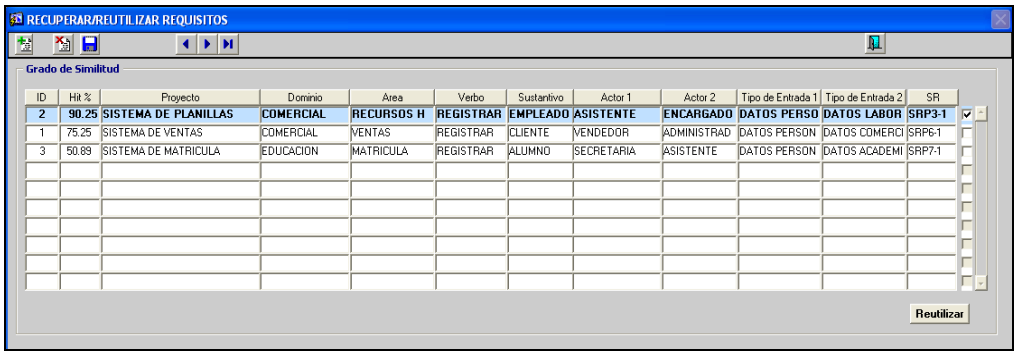


Figura. 6.6. Casos similares del requisito *R1* del caso de estudio 1 utilizando el modelo CBR propuesto.

Como se puede observar en la Figura 6.6, el requisito que se asemeja mejor a nuestro problema es el que tiene ID = 2. Puesto que su grado de similitud es el mejor (90.54%), por ello es escogido para ser reutilizado.

6.3.3.3 Reutilizar Requisito

En esta fase se escoge el requisito con mejor grado de similitud, presionamos el botón *Reutilizar*, para poder usar la información de este requisito como posible solución, tal como se muestra en la Figura 6.7.

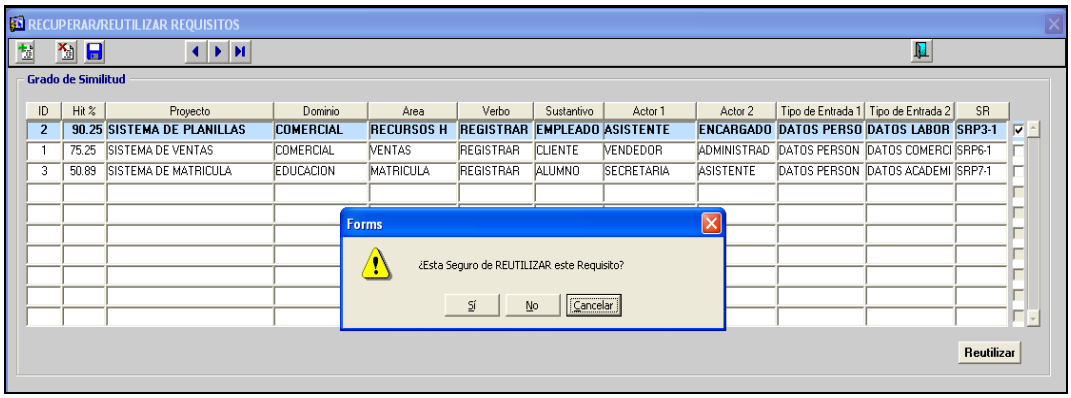


Figura. 6.7. Reutilizar requisito para dar posible solución al requisito R1 del caso de estudio 1 utilizando el modelo CBR propuesto.

Tal como se observa en la Figura 6.8, se crea el nuevo requisito reutilizando los datos del requisito ID=2. Se puede observar que el nuevo requisito tiene ID = 24.

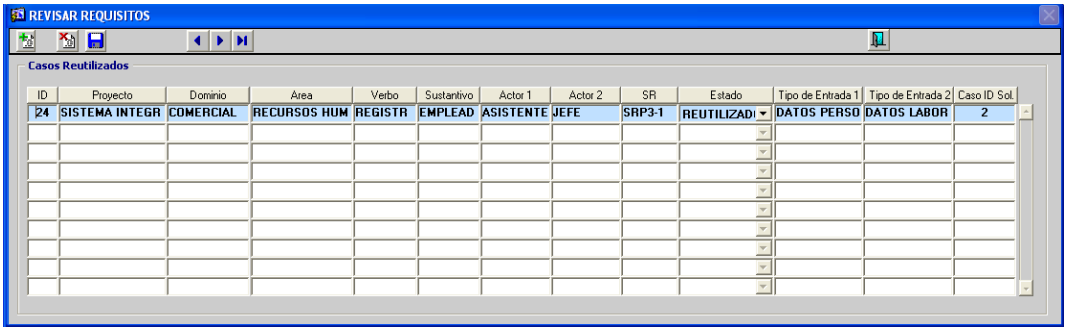


Figura. 6.8. Requisito reutilizado del caso de estudio 1, utilizando el modelo CBR propuesto.

6.3.3.4 Revisar Requisito

Para saber si la solución del requisito es la adecuada, se cambia el estado a *Revisado*. En esta etapa se revisa el documento SR3-1 y se analiza si todo lo especificado allí es correcto.

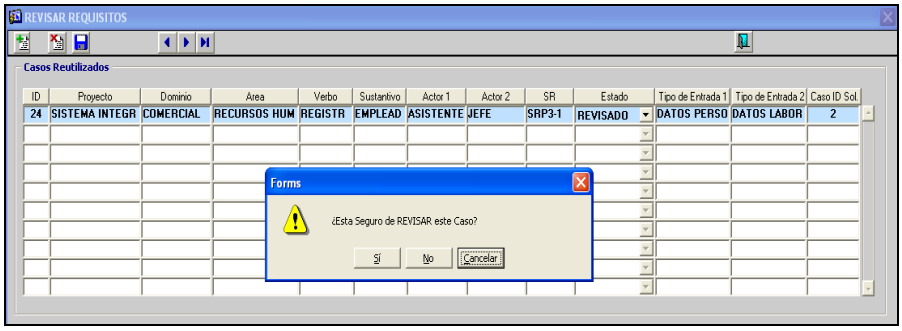


Figura. 6.9. Revisar requisito del caso de estudio 1, utilizando el modelo CBR propuesto.

Estando en esta etapa, el documento SRP3-1 (especificación del requisito) se revisó con el usuario final, donde se realizaron algunos cambios, por ello se cambió el nombre del documento SPR3-1 por SRP3-1C. En el documento SRP3-1C, los cambios que se han realizado están resaltados para una mejor observación.

En los Anexos 6 y 7 se muestran los documentos SRP3-1 y SRP3-1C.

6.3.3.5 Retener Requisito

Como se puede observar en la Figura 6.10, el atributo SR se ha cambiado a SRP3-1C ya que el documento ha tenido algunos pequeños cambios. Finalmente, como se han realizados cambios muy pequeños, se decide en retener el requisito a la LR.

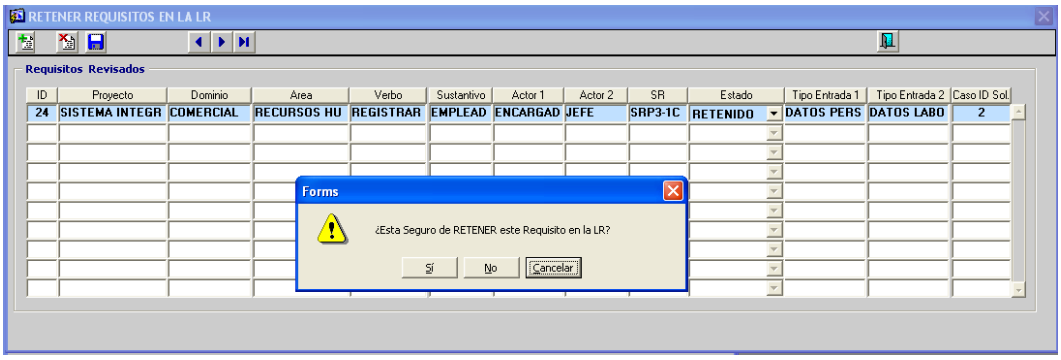


Figura. 6.10. Retener requisito del caso de estudio 1, utilizando el modelo CBR propuesto.

Los resultados de los demás requisitos se muestran en el Anexo 3.

6.4 Pruebas del caso de estudio 2

En el presente caso de estudio se escogieron diez requisitos funcionales del *Sistema de Punto de Venta (SPTA)* para realizar la captación de requisitos, donde también se aplicaron: *Casos de Uso*, *Win Win* y el modelo *CBR* propuesto.

En la Tabla 6.14 se muestra los diez requisitos funcionales del *SPTA* que se han escogido. Como se puede observar, se seleccionaron los requisitos que tienen prioridad *Alta*. La prioridad de los requisitos fue realizada por el usuario final.

Tabla 6.14. Requisitos funcionales prioritarios del Sistema de Punto de Venta (SPTA).

Código	Requisito	Prioridad
R1	Registrar Productos	Alta
R2	Registrar Cliente	Alta
R3	Registrar Venta	Alta
R4	Registrar Compras	Alta
R5	Generar Comprobante	Alta
R6	Aperturar Caja	Alta
R7	Consultar Productos	Alta
R8	Consultar Ventas	Alta
R9	Consultar Stock	Alta
R10	Consultar Kárdex	Alta

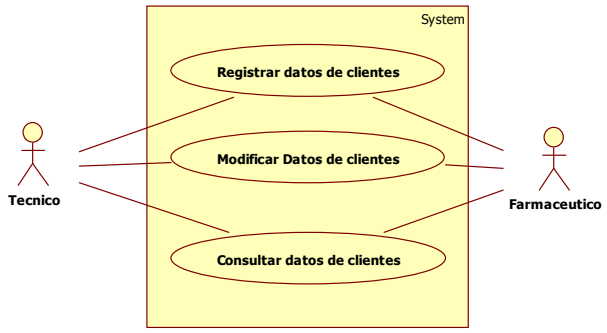
6.4.1 Captación de Requisitos aplicando Casos de Uso

En esta sección se va capturar los requisitos de este caso de estudio, aplicando la técnica *Casos de Usos* explicados en el capítulo 3. La Tabla 6.15 muestra el resumen del *Caso de Uso* del requisito*: R1-Registrar Cliente.* Allí se detalla el diagrama de Casos de Uso, una breve descripción del caso de uso y los actores que intervienen.

Tabla 6.15. *Caso de Uso* del requisito *R2-Registrar Cliente* del caso de estudio 2.

Caso de Uso: R2-Registrar Cliente	**Analista:** Iesus Delgado

Diagrama de Casos de Uso



Breve Descripción

- Este Caso de Uso se refiere a las actividades llevadas a cabo para completar satisfactoriamente el registro de los datos de los clientes, el cual puede ser realizado tanto por el técnico como por el Farmacéutico de cada local.
- El Encargado de personal podrá registrar los datos de los clientes, actualizar y consultar su información.

Actores

- Farmacéutico: usuario interno a la organización que está encargado de dar asesoría relacionada a alguna posología o prescripción médica de los productos farmacéuticos a los clientes. También está destinado a realizar las ventas de los productos farmacéuticos, ya sea en el local o por Delivery. Así mismo, está destinado a dar información del producto, solicitada por los clientes.
- Técnico: usuario interno a la organización que está encargado de realizar las ventas de los productos farmacéuticos, ya sea en el local o por Delivery. También está destinado a dar información del producto, solicitada por los clientes.

Flujo básico

- El sistema muestra la pantalla para acceder al Sistema.
- El actor ingresa los datos de autenticación para ingresar al sistema.
- El sistema inicia la sesión del actor y le presenta la pantalla de Registro de Cliente.

- El actor procede con el llenado de los campos solicitados por el Sistema, los mismos que deberá llenar antes de proceder con los demás pasos. La información que deberá incluir en este rubro consiste en: Tipo de documento de Identidad, Número de documento de identidad, Razón social o Apellido Paterno, Apellido Materno, Nombres, Nacionalidad, Estado Civil, Sexo (Género), Teléfono, Fax, Email, Dirección, etc.
- El actor presiona el botón "Guardar" para guardar los datos ingresados.
- El caso de uso termina

Flujo Alternativo

- El actor decide salir del sistema sin registrar sus datos:
- El actor no ingresa ningún dato y sale de la página donde se encuentra.
- El caso de uso termina.

En la especificación del requisito (SR) se encuentran todos los casos de usos desarrollados. El resumen de los Casos de Usos de los demás requisitos se muestra en el Anexo 4.

6.4.2 Captación de Requisitos aplicando Win Win

En esta sección se va capturar los requisitos de este caso de estudio, aplicando la técnica *Win Win*, explicados en el capítulo 3. Se ha realizado la captación de requisitos en tres iteraciones.

- ***Iteración 1***

En esta fase se tuvo una primera reunión de trabajo con el Jefe del proyecto, los analistas y stackeholders. En esta reunión se identificaron los requisitos que se muestra en la Tabla 6.16.

Tabla 6.16. Requisitos identificados en la iteración 1 utilizando la técnica Win Win aplicada en el caso de estudio 2.

No.	Requisito	Solicitado por	Cargo	Área	Analista responsable	Estado
1	Registrar Productos	Psalcedo	Asistente	Ventas	Idelgado	Propuesto
2	Registrar Cliente	Psalcedo	Asistente	Ventas	Idelgado	Propuesto

3	Registrar Empleado	Psalcedo	Asistente	Ventas	Idelgado	Propuesto
4	Registrar Venta	kcalderon	Coordinador	Ventas	Idelgado	Propuesto
5	Registrar Compras	kcalderon	Coordinador	Ventas	Idelgado	Propuesto
6	Generar Comprobante	kcalderon	Coordinador	Ventas	Idelgado	Propuesto
7	Aperturar Caja	kcalderon	Coordinador	Ventas	Idelgado	Propuesto
8	Consultar Productos	Psalcedo	Asistente	Ventas	Idelgado	Propuesto
9	Consultar Clientes	Psalcedo	Asistente	Ventas	Idelgado	Propuesto
10	Consultar Ventas	kcalderon	Asistente	Ventas	Idelgado	Propuesto
11	Consultar Empleados	Psalcedo	Asistente	Ventas	Idelgado	Propuesto
12	Consultar Stock	kcalderon	Coordinador	Ventas	Cmorales	Propuesto
13	Consultar Kárdex	kcalderon	Coordinador	Ventas	Cmorales	Propuesto
14	Carga Inicial de Inventario	kcalderon	Coordinador	Ventas	Cmorales	Propuesto
15	Registrar Inventario	kcalderon	Coordinador	Ventas	Cmorales	Propuesto
16	Registrar guías de almacén	kcalderon	Coordinador	Ventas	Cmorales	Propuesto
17	Consultar guías	kcalderon	Coordinador	Ventas	Cmorales	Propuesto
18	Registrar Proveedores	Psalcedo	Asistente	Ventas	Idelgado	Propuesto
19	Registra Traspasos de mercaderías	kcalderon	Coordinador	Ventas	Cmorales	Propuesto
20	Recepción de pedidos Delivery	kcalderon	Coordinador	Ventas	Idelgado	Propuesto

- ***Requisitos identificados en la Iteración 2***

En esta segunda fase se tuvo una segunda reunión de trabajo, donde se especificaron con más detalle los requisitos. Los stakeholders o interesados dieron prioridades a los requisitos y, de acuerdo a estas prioridades, se aceptaron o rechazaron los requisitos (ver Tabla 6.17).

Tabla 6.17. Requisitos identificados en la iteración 2 utilizando la técnica Win Win aplicada en el caso de estudio 2.

No.	Requisito	Descripción	Estado	Prioridad
1	Registrar Productos	El sistema deberá permitir registrar los datos de los productos farmacéuticos, tales como: códigos, descripción resumida del producto, descripción detallada, familia, subfamilia, estado, precio de compra, precio de venta, posología, etc.	Aceptado	Alta

2	Registrar Cliente	El sistema deberá permitir registrar los datos de los clientes, tales como: nombres y apellidos o razón social, tipo de documento de identidad, número de documento de identidad, número telefónico, dirección, correo electrónico, etc.	Aceptado	Alta
3	Registrar Empleado	El sistema deberá permitir registrar los datos de los empleados que laboran en la empresa, tales como: nombres y apellidos, tipo de documento de identidad, número de documento de identidad, número telefónico, dirección, correo electrónico, etc.	Aceptado	Media
4	Registrar Venta	El sistema deberá permitir registrar las ventas realizadas por cada uno de los diferentes locales. Se debe registrar los datos generales y el detalle de las ventas, tales como: fecha, tipo de comprobante, número de comprobante, importe total, importe del IGV, cliente, productos, cantidad, precio por productos, etc. Estas ventas pueden ser hechas en el local o por Delivery.	Aceptado	Alta
5	Registrar Compras	El sistema deberá permitir registrar las compras que la empresa realiza. Se debe registrar los datos generales y el detalle de las compras, tales como: tipo de comprobante, número de comprobante, fecha, importe total, importe del IGV, proveedor, productos, cantidad, precio por productos, etc.	Aceptado	Alta
6	Generar Comprobante	El sistema deberá permitir generar el comprobante de pago luego de haber realizado la venta. Este comprobante también debe ser impreso.	Aceptado	Alta
7	Aperturar Caja	El sistema deberá permitir realizar la apertura del día para realizar las ventas.	Aceptado	Alta
8	Consultar Productos	El sistema deberá permitir consultar los datos de los productos, tales como: descripción resumida del producto, descripción detallada, familia, subfamilia, estado, precio de compra, precio de venta, posología, etc.	Aceptado	Alta
9	Consultar Clientes	El sistema deberá permitir consultar los datos de los clientes, tales como: nombres y apellidos o razón social, tipo de documento de identidad, número de documento de identidad, número telefónico, dirección, correo electrónico, etc.	Aceptado	Media
10	Consultar Ventas	El sistema deberá permitir consultar las ventas realizadas por cada uno de los diferentes locales. Se deberá permitir consultar las ventas por local, por mes, por año, por rango de fechas, etc.	Aceptado	Alta
11	Consultar Empleados	El sistema deberá permitir consultar los datos de los empleados, tales como: nombres y apellidos, número de documento de identidad, número telefónico, dirección, etc.	Aceptado	Media
12	Consultar Stock	El sistema deberá permitir consultar el stock actual de un determinado producto de un determinado local.	Aceptado	Alta
13	Consultar Kárdex	El sistema deberá permitir consultar el kárdex de un determinado producto por local.	Aceptado	Alta
14	Carga Inicial de Inventario	El sistema deberá permitir generar una carga inicial del inventario de los productos, por local.	Aceptado	Media

15	Registrar Inventario	El sistema deberá permitir registrar los inventarios que se realicen en cada local, en una determinada fecha.	Aceptado	Media
16	Registrar guías de almacén	El sistema deberá permitir registrar las guías de las salidas de mercaderías del almacén. Se debe registrar los datos generales y el detalle de las guías, tales como: fecha, número, productos, cantidad, etc.	Aceptado	Media
17	Consultar guías	El sistema deberá permitir consultar las guías de cada local.	Aceptado	Media
18	Registrar proveedores	El sistema deberá permitir registrar los datos de los proveedores, tales como: nombres y apellidos o razón social, tipo de documento de identidad, número de documento de identidad, número telefónico, etc.	Aceptado	Baja
19	Registra Traspasos de mercaderías	El sistema deberá permitir registrar los traspasos de mercaderías que se realizan. Se debe registrar el origen y destino de la mercadería.	Aceptado	Media
20	Recepción de pedidos Delivery	El sistema deberá permitir registrar los pedidos que se realizan para las ventas por Delivery. Se deberá registrar: datos del cliente, datos del productos, cantidad, etc.	Aceptado	Media

- ***Requisitos identificados en la Iteración 3***

En esta tercera fase se escogieron los requisitos que tienen prioridad *Alta*. Luego se validaron cada uno de ellos y se realizaron los ajustes necesarios (ver Tabla 6.18).

Tabla 6.18. Requisitos identificados en la iteración 3 utilizando la técnica Win Win aplicada en el caso de estudio 2.

No.	Requisito	Solicitado por	Analista responsable	Validado Por	Estado	Prioridad
1	Registrar Productos	Psalcedo	Eiparraguirre	Idelgado	Validado	Alta
2	Registrar Cliente	Psalcedo	Eiparraguirre	Idelgado	Validado	Alta
4	Registrar Venta	kcalderon	Eiparraguirre	Idelgado	Validado	Alta
5	Registrar Compras	kcalderon	Eiparraguirre	Idelgado	Validado	Alta
6	Generar Comprobante	kcalderon	Eiparraguirre	Idelgado	Validado	Alta
7	Aperturar Caja	kcalderon	Eiparraguirre	Idelgado	Validado	Alta
8	Consultar Productos	Psalcedo	Eiparraguirre	Idelgado	Validado	Alta
10	Consultar Ventas	kcalderon	Eiparraguirre	Idelgado	Validado	Alta
12	Consultar Stock	kcalderon	Eiparraguirre	Cmorales	Validado	Alta
13	Consultar Kárdex	kcalderon	Eiparraguirre	Cmorales	Validado	Alta

6.4.3 Captación de Requisitos aplicando el modelo CBR propuesto

En esta sección se realiza la captación de requisitos del mismo caso de estudio, pero aplicando el *modelo CBR propuesto*. A continuación se muestra cada una de las tareas del ciclo de vida CBR de nuestro modelo.

6.4.3.1 Recuperar Requisito

En la Figura 6.10 se muestra cómo se ha recuperado el requisito *R2.Registrar Cliente* del presente caso de estudio. Para ello se tiene que ingresar todos los atributos del nuevo requisito que se quiere resolver. Además, se muestra todos los atributos, tales como: nombre del proyecto, el dominio del proyecto, área, el verbo y el sustantivo del requisito, etc.

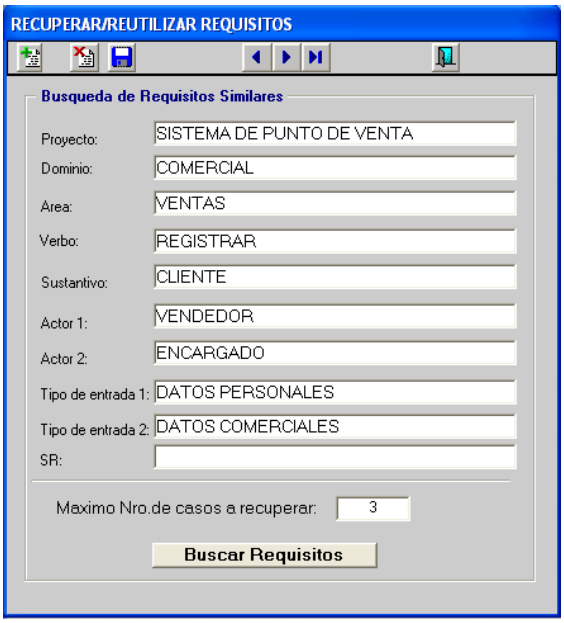
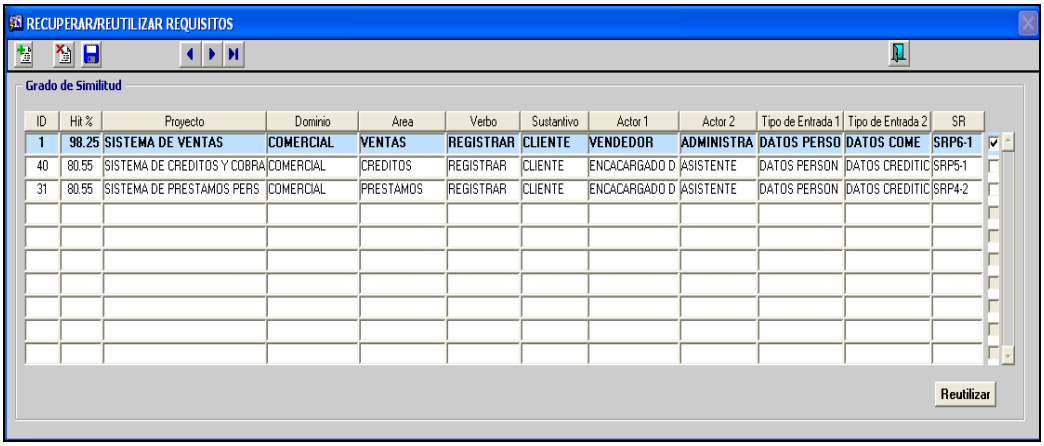


Figura. 6.11. Recuperación del requisito *R2* del caso de estudio 2 utilizando el modelo CBR propuesto.

Después de ingresar los valores de los atributos, se procede a ingresar el número máximo de casos que se quiere recuperar, tal como se muestra en la Figura 6.10, donde se desea que se muestre como máximo tres casos similares a nuestro requisito.

A continuación se muestran los tres requisitos que tienen el mejor grado de similitud (ver Figura 6.11):



ID	Hit %	Proyecto	Dominio	Area	Verbo	Sustantivo	Actor 1	Actor 2	Tipo de Entrada 1	Tipo de Entrada 2	SR
1	98.25	SISTEMA DE VENTAS	COMERCIAL	VENTAS	REGISTRAR	CLIENTE	VENDEDOR	ADMINISTRA	DATOS PERSO	DATOS COME	SRP6-1
40	80.55	SISTEMA DE CREDITOS Y COBRA	COMERCIAL	CREDITOS	REGISTRAR	CLIENTE	ENCACARGADO D	ASISTENTE	DATOS PERSON	DATOS CREDITIC	SRP5-1
31	80.55	SISTEMA DE PRESTAMOS PERS	COMERCIAL	PRESTAMOS	REGISTRAR	CLIENTE	ENCACARGADO D	ASISTENTE	DATOS PERSON	DATOS CREDITIC	SRP4-2

Figura. 6.12. Casos similares del requisito *R2* del caso de estudio 2 utilizando el modelo CBR propuesto.

Como se puede observar en la Figura 6.11, el requisito que se asemeja mejor a nuestro problema es el que tiene ID = 1. Puesto que su grado de similitud es el mejor (98.25%), por ello es escogido para ser reutilizado.

6.4.3.2 Reutilizar Requisito

En esta fase se escoge el requisito con mejor grado de similitud, presionamos el botón *Reutilizar* para poder usar la información de este requisito como posible solución, tal como se muestra en la Figura 6.12.

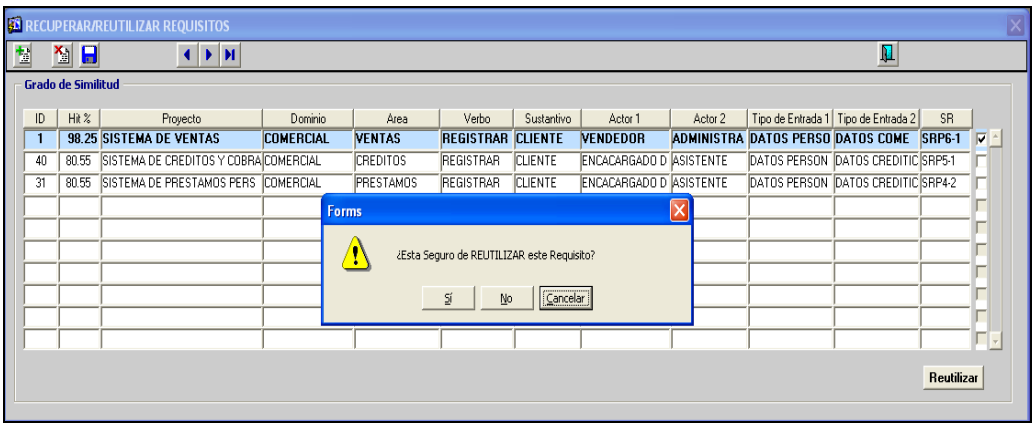


Figura. 6.13. Reutilizar requisito para dar posible solución al requisito R2 del caso de estudio 2 utilizando el modelo CBR propuesto.

Tal como se observa en la Figura 6.13, se crea el nuevo requisito reutilizando los datos del requisito ID=1. Se puede observar que el nuevo requisito tiene ID = 56.

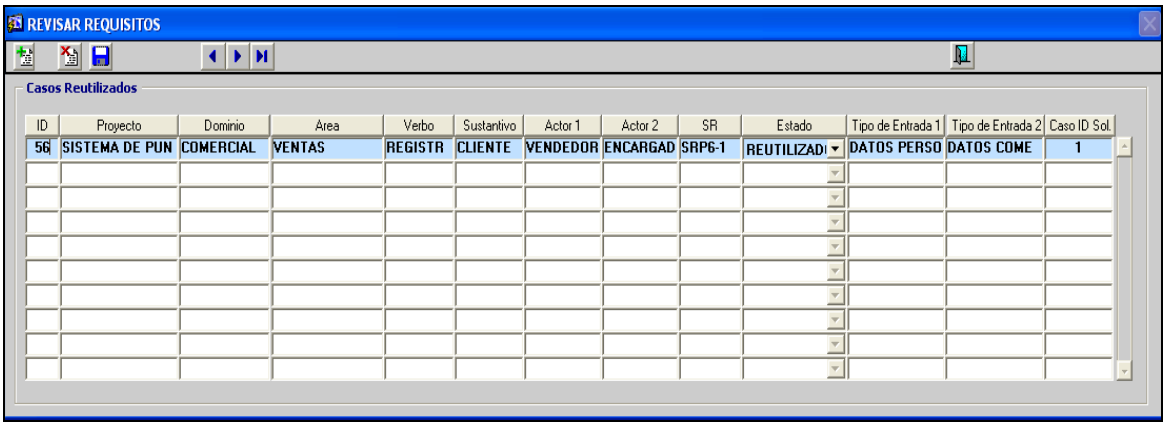


Figura. 6.14. Requisito reutilizado del caso de estudio 2, utilizando el modelo CBR propuesto.

6.4.3.3 Revisar Requisito

Para saber si la solución de requisito es la adecuada, se cambia el estado a *Revisado*. En esta etapa se revisa el documento SRP6-1 con el usuario final.

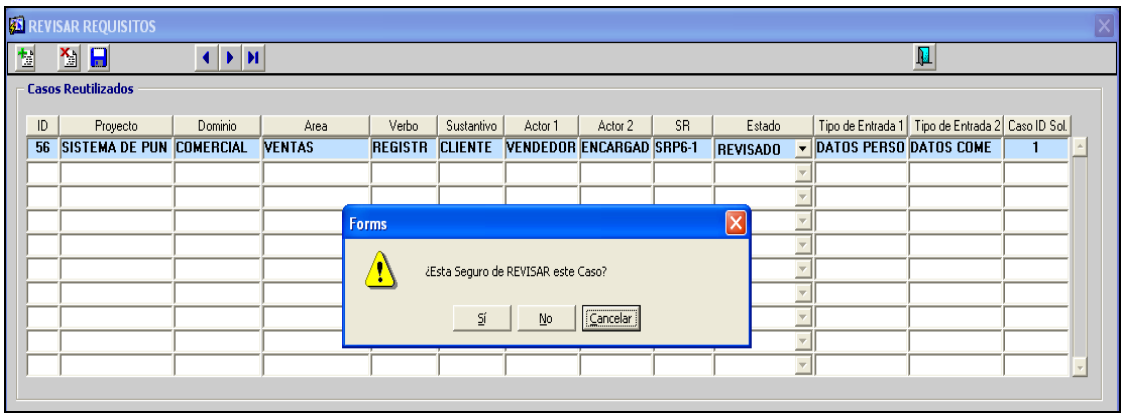


Figura. 6.15. Revisar requisito del caso de estudio 2, utilizando el modelo CBR propuesto.

El documento SRP6-1 (especificación del requisito) se revisó con el usuario final. Todo lo especificado allí es correcto, por ello no se cambia el nombre del documento (ver Figura 6.15).

6.4.3.4 Retener Requisito

Finalmente, como todo está conforme, se cambia el estado del requisito a *Retenido* (ver Figura 6.16).

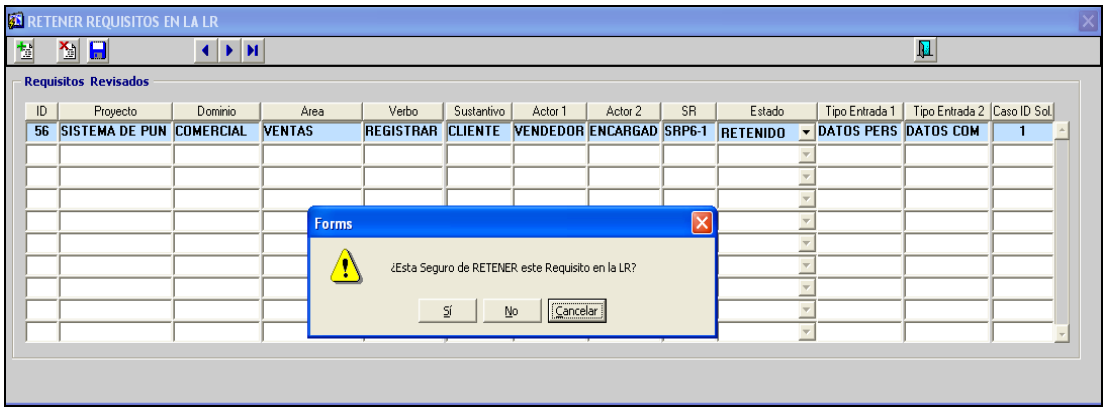


Figura. 6.16. Retener requisito del caso de estudio 2, utilizando el modelo CBR propuesto.

Los resultados de los demás requisitos se muestran en el Anexo 5.

6.5 Análisis de pruebas y resultados

En esta sección se va aplicar métricas de calidad, definidas en la sección 6.2, a los requisitos obtenidos con el *Modelo CBR* propuesto de cada caso de estudio. Luego se realizará una comparación del tiempo invertido en la captación de Requisitos utilizando *Casos de Uso*, *Win Win* y el *Modelo CBR* propuesto.

6.5.1 Uso de Métricas de Calidad en el caso de estudio 1: Sistema Integrado de Recursos Humanos (SIRH)

A continuación se aplicará las métricas de calidad definidos en la sección 6.2, a los requisitos obtenidos con el *Modelo CBR propuesto*, *Casos de Uso* y *Win Win* del caso de estudio 1*: Sistema Integrado de Recursos Humanos (SIRH).*

6.5.1.1 Requisitos obtenidos aplicando el Modelo CBR propuesto

- **Métrica de calidad: Claridad de descripción**

Aplicando la métrica de calidad MC001, explicado en la sección 6.2.1, se obtiene el resultado que se muestra en la Tabla 6.19.

Tabla 6.19. Resultados obtenidos al aplicar la métrica de calidad MC001 a los requisitos obtenidos con el modelo *CBR* propuesto del caso de estudio 1.

Requisitos	Requisito entendido 100%	Requisitos No entendido 100%
R1. Registrar Empleado	√	
R2. Aperturar Planilla	√	
R3. Cerrar Planilla	√	
R4. Generar Planilla	√	
R5. Generar Boleta	√	
R6. Generar Liquidaciones	√	
R7. Registrar Licencias	√	
R8. Registrar Vacaciones	√	
R9. Imprimir Planilla	√	
R10.Generar PDT	√	
Total:	10	0
Métrica: A/B = 10/10 = 1 Donde: A = Nro. de requisitos entendidos B = Nro. total de requisitos		

En la Tabla 6.19 se puede observar que el resultado de la métrica es 1, esto quiere decir que todos los requisitos han sido entendidos en un 100%.

- **Métrica de calidad: Completitud funcional**

Aplicando la métrica de calidad MC002, explicado en la sección 6.2.2, se obtiene el resultado que se muestra en la Tabla 6.20.

Tabla 6.20. Resultados obtenidos al aplicar la métrica de calidad MC002 a los requisitos obtenidos con el modelo *CBR* propuesto del caso de estudio 1.

Requisitos	Requisito Satisface 100%	Requisitos No Satisfacen 100%
R1. Registrar Empleado	√	
R2. Aperturar Planilla	√	
R3. Cerrar Planilla	√	
R4. Generar Planilla	√	
R5. Generar Boleta	√	
R6. Generar Liquidaciones	√	
R7. Registrar Licencias	√	
R8. Registrar Vacaciones	√	
R9. Imprimir Planilla	√	
R10.Generar PDT	√	
Total:	10	0

Métrica: A/B = 10/10 = 1

Donde:

A = Nro. de requisitos satisfacen 100% las necesidades del usuario

B = Nro. total de requisitos

En la Tabla 6.20 se puede observar que el resultado de la métrica es 1, esto quiere decir que los requisitos satisfacen en un 100% las necesidades del usuario.

- **Métrica de calidad: Reusabilidad**

Aplicando la métrica de calidad MC003, explicado en la sección 6.2.3, se obtiene el resultado que se muestra en la Tabla 6.21.

Tabla 6.21. Resultados obtenidos al aplicar la métrica de calidad MC003 a los requisitos obtenidos con el modelo *CBR* propuesto del caso de estudio 1.

Requisitos	Requisitos Reusados	Requisitos no Reusados
R1. Registrar Empleado	√	
R2. Aperturar Planilla	√	

R3. Cerrar Planilla	√	
R4. Generar Planilla	√	
R5. Generar Boleta	√	
R6. Generar Liquidaciones		√
R7. Registrar Licencias		√
R8. Registrar Vacaciones	√	
R9. Imprimir Planilla	√	
R10.Generar PDT	√	
Total:	8	2

Métrica: A/B = 8/10 = 0.8
Donde:
A = Nro. de requisitos reusados para identificar otros requisitos
B = Nro. total de requisitos

En la Tabla 6.21 se puede observar que el resultado de la métrica es 0.8, esto quiere decir que los requisitos son reusados en un 80 % para capturar requisitos en nuevos proyectos.

- **Métrica de calidad: Eficiencia**

En la Tabla 6.22 se muestra el tiempo invertido en cada una de las tareas del modelo propuesto del requisito R1. Registrar Empleado.

Tabla 6.22. Tiempo invertido en la captación del requisito *R1.Registrar Empleado* con el modelo *CBR* propuesto del caso de estudio 1.

Tareas con CBR	Tiempo (*)
Recuperar Requisitos	2
Reutilizar	1
Revisar	20
Modificar SR	1
Retener	2
Total:	26
(*) Tiempo en minutos	

6.5.1.2 Requisitos obtenidos aplicando Casos de Uso

- **Métrica de calidad: Claridad de descripción**

Aplicando la métrica de calidad MC001, explicado en la sección 6.2.1, se obtiene el resultado que se muestra en la Tabla 6.23.

Tabla 6.23. Resultados obtenidos al aplicar la métrica de calidad MC001 a los requisitos obtenidos aplicando Caso de Usos del caso de estudio 1.

Requisitos	Requisito entendido 100%	Requisitos No entendido 100%
R1. Registrar Empleado	√	
R2. Aperturar Planilla	√	
R3. Cerrar Planilla	√	
R4. Generar Planilla	√	
R5. Generar Boleta	√	
R6. Generar Liquidaciones	√	
R7. Registrar Licencias	√	
R8. Registrar Vacaciones	√	
R9. Imprimir Planilla	√	
R10.Generar PDT	√	
Total:	10	0
Métrica: A/B = 10/10 = 1 Donde: A = Nro. de requisitos entendidos B = Nro. total de requisitos		

En la Tabla 6.23 se puede observar que el resultado de la métrica es 1, esto quiere decir que todos los requisitos han sido entendidos en un 100%.

- **Métrica de calidad: Completitud funcional**

Aplicando la métrica de calidad MC002, explicado en la sección 6.2.2, se obtiene el resultado que se muestra en la Tabla 6.24.

Tabla 6.24. Resultados obtenidos al aplicar la métrica de calidad MC002 a los requisitos obtenidos aplicando Caso de Usos del caso de estudio 1.

Requisitos	Requisito Satisface 100%	Requisitos No Satisfacen 100%
R1. Registrar Empleado	√	
R2. Aperturar Planilla	√	
R3. Cerrar Planilla	√	
R4. Generar Planilla	√	
R5. Generar Boleta	√	
R6. Generar Liquidaciones	√	
R7. Registrar Licencias	√	
R8. Registrar Vacaciones	√	
R9. Imprimir Planilla	√	
R10.Generar PDT		√
Total:	9	1
Métrica: A/B = 9/10 = 0.9 Donde: A = Nro. de requisitos satisfacen 100% las necesidades del usuario B = Nro. total de requisitos		

En la Tabla 6.24 se puede observar que el resultado de la métrica es 0.9, esto quiere decir que los requisitos satisfacen en un 90% las necesidades del usuario.

- **Métrica de calidad: Reusabilidad**

Aplicando la métrica de calidad MC003, explicado en la sección 6.2.3, se obtiene el resultado que se muestra en la Tabla 6.25.

Tabla 6.25. Resultados obtenidos al aplicar la métrica de calidad MC003 a los requisitos obtenidos aplicando Caso de Usos del caso de estudio 1.

Requisitos	Requisitos Reusados	Requisitos no Reusados
R1. Registrar Empleado	√	
R2. Aperturar Planilla		√

R3. Cerrar Planilla		√
R4. Generar Planilla		√
R5. Generar Boleta		√
R6. Generar Liquidaciones		√
R7. Registrar Licencias		√
R8. Registrar Vacaciones		√
R9. Imprimir Planilla		√
R10.Generar PDT		√
Total:	1	9
Métrica: A/B = 1/10 = 0.1 Donde: A = Nro. de requisitos reusados para identificar otros requisitos B = Nro. total de requisitos		

En la Tabla 6.25 se puede observar que el resultado de la métrica es 0.1, esto quiere decir que los requisitos son reusados en un 10 % para capturar requisitos en nuevos proyectos.

- **Métrica de calidad: Eficiencia**

En la Tabla 6.26 se muestra el tiempo invertido en la identificación de los requisitos del caso de estudio 1, aplicando la técnica Win Win.

Tabla 6.26. Tiempo invertido en la captación de requisitos aplicando Casos de Usos del caso de estudio 1.

Requisitos	Tiempo en minutos
Registrar Empleado	105
Aperturar Planilla	45
Cerrar Planilla	45
Generar Planilla	180
Generar Boleta	80
Generar Liquidaciones	120

Registrar Licencias	110
Registrar Vacaciones	120
Imprimir Planilla	95
Generar PDT	120
Totales	**1020**

6.5.1.3 Requisitos obtenidos aplicando Win Win

- **Métrica de calidad: Claridad de descripción**

Aplicando la métrica de calidad MC001, explicado en la sección 6.2.1, se obtiene el resultado que se muestra en la Tabla 6.27.

Tabla 6.27. Resultados obtenidos al aplicar la métrica de calidad MC001 a los requisitos obtenidos aplicando Win Win del caso de estudio 1.

Requisitos	**Requisito entendido 100%**	**Requisitos No entendido 100%**
R1. Registrar Empleado	√	
R2. Aperturar Planilla	√	
R3. Cerrar Planilla	√	
R4. Generar Planilla	√	
R5. Generar Boleta	√	
R6. Generar Liquidaciones	√	
R7. Registrar Licencias	√	
R8. Registrar Vacaciones	√	
R9. Imprimir Planilla	√	
R10.Generar PDT	√	
Total:	10	0
Métrica: A/B = 10/10 = 1 Donde: A = Nro. de requisitos entendidos B = Nro. total de requisitos		

En la Tabla 6.27 se puede observar que el resultado de la métrica es 1, esto quiere decir que todos los requisitos han sido entendidos en un 100%.

- **Métrica de calidad: Completitud funcional**

Aplicando la métrica de calidad MC002, explicado en la sección 6.2.2, se obtiene el resultado que se muestra en la Tabla 6.28.

Tabla 6.28. Resultados obtenidos al aplicar la métrica de calidad MC002 a los requisitos obtenidos aplicando Caso de Usos del caso de estudio 1.

Requisitos	Requisito Satisface 100%	Requisitos No Satisfacen 100%
R1. Registrar Empleado	√	
R2. Aperturar Planilla	√	
R3. Cerrar Planilla	√	
R4. Generar Planilla	√	
R5. Generar Boleta	√	
R6. Generar Liquidaciones		√
R7. Registrar Licencias	√	
R8. Registrar Vacaciones	√	
R9. Imprimir Planilla	√	
R10.Generar PDT	√	
Total:	9	1

Métrica: A/B = 9/10 = 0.9
Donde:
A = Nro. de requisitos satisfacen 100% las necesidades del usuario
B = Nro. total de requisitos

En la Tabla 6.28 se puede observar que el resultado de la métrica es 0.9, esto quiere decir que los requisitos satisfacen en un 90% las necesidades del usuario.

- **Métrica de calidad: Reusabilidad**

Aplicando la métrica de calidad MC003, explicado en la sección 6.2.3, se obtiene el resultado que se muestra en la Tabla 6.29.

Tabla 6.29. Resultados obtenidos al aplicar la métrica de calidad MC003 a los requisitos obtenidos aplicando Win Win del caso de estudio 1.

Requisitos	Requisitos Reusados	Requisitos no Reusados
R1. Registrar Empleado	√	
R2. Aperturar Planilla		√
R3. Cerrar Planilla		√
R4. Generar Planilla		√
R5. Generar Boleta		√
R6. Generar Liquidaciones		√
R7. Registrar Licencias		√
R8. Registrar Vacaciones		√
R9. Imprimir Planilla		√
R10.Generar PDT		√
Total:	1	9

Métrica: A/B = 1/10 = 0.1
Donde:
A = Nro. de requisitos reusados para identificar otros requisitos
B = Nro. total de requisitos

En la Tabla 6.29 se puede observar que el resultado de la métrica es 0.1, esto quiere decir que los requisitos son reusados en un 10 % para capturar requisitos en nuevos proyectos.

- **Métrica de calidad: Eficiencia**

En la Tabla 6.30 se muestra el tiempo invertido en la identificación de los requisitos del caso de estudio 1, aplicando la técnica Win Win.

Tabla 6.30. Tiempo invertido en la captación de requisitos aplicando Win Win del caso de estudio 1.

Requisitos	Win Win
Registrar empleado	90

Aperturar Planilla	35
Cerrar Planilla	40
Generar Planilla	150
Generar Boleta	60
Generar Liquidaciones	100
Registrar Licencias	90
Registrar Vacaciones	100
Imprimir Planilla	75
Generar PDT	100
Totales	**840**

6.5.2 Uso de Métricas de Calidad en el caso de estudio 2

A continuación se aplicará las métricas de calidad definidos en la sección 6.2, a los requisitos obtenidos con el *Modelo CBR propuesto*, *Casos de Uso* y *Win Win* del caso de estudio 2: *Sistema de Punto de Venta (SPTA).*

6.5.2.1 Requisitos obtenidos aplicando el Modelo CBR propuesto

- **Métrica de calidad: Claridad de descripción**

Aplicando la métrica de calidad MC001, explicado en la sección 6.2.1, se obtiene el resultado que se muestra en la Tabla 6.31.

Tabla 6.31. Resultados obtenidos al aplicar la métrica de calidad MC001 a los requisitos obtenidos con el modelo *CBR* propuesto del caso de estudio 2.

Requisitos	Requisito entendido 100%	Requisitos No entendido 100%
R1. Registrar Productos	√	
R2. Registrar Cliente	√	
R3. Registrar Venta	√	
R4. Registrar Compras	√	
R5. Generar Comprobante	√	

R6. Aperturar Caja	√	
R7. Consultar Productos	√	
R8. Consultar Ventas	√	
R9. Consultar Stock	√	
R10.Consultar Kárdex	√	
Total:	10	0
Métrica: A/B = 10/10 = 1 Donde: A = Nro. de requisitos entendidos B = Nro. total de requisitos		

En la Tabla 6.31 se puede observar que el resultado de la métrica es 1.00, esto quiere decir que todos los requisitos se han sido entendidos en un 100%.

- **Métrica de calidad: Completitud funcional**

Aplicando la métrica de calidad MC002, explicado en la sección 6.2.2, se obtiene el resultado que se muestra en la Tabla 6.32.

Tabla 6.32. Resultados obtenidos al aplicar la métrica de calidad MC002 a los requisitos obtenidos con el modelo *CBR* propuesto del caso de estudio 2.

Requisitos	Requisito Satisface 100%	Requisitos No Satisfacen 100%
R1. Registrar Productos	√	
R2. Registrar Cliente	√	
R3. Registrar Venta	√	
R4. Registrar Compras	√	
R5. Generar Comprobante	√	
R6. Aperturar Caja	√	
R7. Consultar Productos	√	
R8. Consultar Ventas	√	
R9. Consultar Stock	√	
R10.Consultar Kárdex	√	
Total:	10	0

Métrica:	A/B = 10/10 = 1
Donde:	
A = Nro. de requisitos satisfacen 100% las necesidades del usuario	
B = Nro. total de requisitos	

En la Tabla 6.32 se puede observar que el resultado de la métrica es 1, esto quiere decir que los requisitos satisfacen en un 100% las necesidades del usuario.

- **Métrica de calidad: Reusabilidad**

Aplicando la métrica de calidad MC003, explicado en la sección 6.2.3, se obtiene el resultado que se muestra en la Tabla 6.33.

Tabla 6.33. Resultados obtenidos al aplicar la métrica de calidad MC003 a los requisitos obtenidos con el modelo *CBR* propuesto del caso de estudio 2.

Requisitos	Requisitos Reusados	Requisitos no Reusados
R1. Registrar Productos	√	
R2. Registrar Cliente	√	
R3. Registrar Venta	√	
R4. Registrar Compras	√	
R5. Generar Comprobante	√	
R6. Aperturar Caja	√	
R7. Consultar Productos	√	
R8. Consultar Ventas	√	
R9. Consultar Stock		√
R10.Consultar Kárdex		√
Total:	8	2
Métrica: A/B = 8/10 = 0.8 Donde: A = Nro. de requisitos reusados para identificar otros requisitos B = Nro. total de requisitos		

En la Tabla 6.33 se puede observar que el resultado de la métrica es 0.8, esto quiere decir que los requisitos son reusados en un 80 % para capturar requisitos en nuevos proyectos.

- **Métrica de calidad: Eficiencia**

En la Tabla 6.34 se muestra el tiempo invertido en cada una de las tareas del modelo propuesto del requisito *R2. Registrar Cliente*.

Tabla 6.34. Tiempo invertido en la captación del requisito *R2.Registrar Cliente* con el modelo *CBR* propuesto del caso de estudio 2.

Tareas con CBR	Tiempo (*)
Recuperar Requisitos	2
Reutilizar	1
Revisar	17
Modificar SR	1
Retener	2
Total:	23
(*) Tiempo en minutos	

6.5.2.2 Requisitos obtenidos aplicando Casos de Uso

- **Métrica de calidad: Claridad de descripción**

Aplicando la métrica de calidad MC001, explicado en la sección 6.2.1, se obtiene el resultado que se muestra en la Tabla 6.35.

Tabla 6.35. Resultados obtenidos al aplicar la métrica de calidad MC001 a los requisitos obtenidos aplicando Casos de Usos del caso de estudio 2.

Requisitos	Requisito entendido 100%	Requisitos No entendido 100%
R1. Registrar Productos	√	

R2. Registrar Cliente	√	
R3. Registrar Venta	√	
R4. Registrar Compras	√	
R5. Generar Comprobante	√	
R6. Aperturar Caja	√	
R7. Consultar Productos	√	
R8. Consultar Ventas	√	
R9. Consultar Stock	√	
R10.Consultar Kárdex	√	
Total:	10	0

Métrica: A/B = 10/10 = 1
Donde:
A = Nro. de requisitos entendidos 100%
B = Nro. total de requisitos

En la Tabla 6.35 se puede observar que el resultado de la métrica es 1.00, esto quiere decir que todos los requisitos se han sido entendidos en un 100%.

- **Métrica de calidad: Completitud funcional**

Aplicando la métrica de calidad MC002, explicado en la sección 6.2.2, se obtiene el resultado que se muestra en la Tabla 6.36.

Tabla 6.36. Resultados obtenidos al aplicar la métrica de calidad MC002 a los requisitos obtenidos aplicando Casos de Usos del caso de estudio 2.

Requisitos	Requisito Satisface 100%	Requisitos No Satisfacen 100%
R1. Registrar Productos	√	
R2. Registrar Cliente	√	
R3. Registrar Venta	√	
R4. Registrar Compras	√	
R5. Generar Comprobante	√	
R6. Aperturar Caja	√	
R7. Consultar Productos	√	

R8. Consultar Ventas	√	
R9. Consultar Stock	√	
R10.Consultar Kárdex		√
Total:	9	1

Métrica: A/B = 9/10 = 0.9
Donde:
A = Nro. de requisitos satisfacen 100% las necesidades del usuario
B = Nro. total de requisitos

En la Tabla 6.36 se puede observar que el resultado de la métrica es 0.9, esto quiere decir que los requisitos satisfacen en un 90% las necesidades del usuario.

- **Métrica de calidad: Reusabilidad**

Aplicando la métrica de calidad MC003, explicado en la sección 6.2.3, se obtiene el resultado que se muestra en la Tabla 6.37.

Tabla 6.37. Resultados obtenidos al aplicar la métrica de calidad MC003 a los requisitos obtenidos aplicando Casos de Usos del caso de estudio 2.

Requisitos	Requisitos Reusados	Requisitos no Reusados
R1. Registrar Productos	√	
R2. Registrar Cliente	√	
R3. Registrar Venta		√
R4. Registrar Compras		√
R5. Generar Comprobante		√
R6. Aperturar Caja		√
R7. Consultar Productos		√
R8. Consultar Ventas		√
R9. Consultar Stock		√
R10.Consultar Kárdex		√
Total:	2	8

Métrica: A/B = 2/10 = 0.2
Donde:
A = Nro. de requisitos reusados para identificar otros requisitos
B = Nro. total de requisitos

En la Tabla 6.37 se puede observar que el resultado de la métrica es 0.2, esto quiere decir que los requisitos son reusados en un 20 % para capturar requisitos en nuevos proyectos.

- **Métrica de calidad: Eficiencia**

En la Tabla 6.38 se muestra el tiempo invertido en la identificación de los requisitos del caso de estudio 2, aplicando la técnica Casos de Usos.

Tabla 6.38. Tiempo invertido en la captación de requisitos aplicando Casos de Usos del caso de estudio 2.

Requisitos	Tiempo en minutos
Registrar Productos	125
Registrar Cliente	110
Registrar Venta	130
Registrar Compras	120
Generar Comprobante	80
Aperturar Caja	45
Consultar Productos	60
Consultar Ventas	50
Consultar Stock	60
Consultar Kárdex	65
Totales	**845**

6.5.2.3 Requisitos obtenidos aplicando Win Win

- **Métrica de calidad: Claridad de descripción**

Aplicando la métrica de calidad MC001, explicado en la sección 6.2.1, se obtiene el resultado que se muestra en la Tabla 6.39.

Tabla 6.39. Resultados obtenidos al aplicar la métrica de calidad MC001 a los requisitos obtenidos aplicando Win Win del caso de estudio 2.

Requisitos	Requisito entendido 100%	Requisitos No entendido 100%
R1. Registrar Productos	√	
R2. Registrar Cliente	√	
R3. Registrar Venta	√	
R4. Registrar Compras	√	
R5. Generar Comprobante	√	
R6. Aperturar Caja	√	
R7. Consultar Productos	√	
R8. Consultar Ventas	√	
R9. Consultar Stock	√	
R10.Consultar Kárdex	√	
Total:	10	0
Métrica: A/B = 10/10 = 1 Donde: A = Nro. de requisitos entendidos B = Nro. total de requisitos		

En la Tabla 6.39 se puede observar que el resultado de la métrica es 1.00, esto quiere decir que todos los requisitos se han sido entendidos en un 100%.

- **Métrica de calidad: Completitud funcional**

Aplicando la métrica de calidad MC002, explicado en la sección 6.2.2, se obtiene el resultado que se muestra en la Tabla 6.40.

Tabla 6.40. Resultados obtenidos al aplicar la métrica de calidad MC002 a los requisitos obtenidos aplicando Win Win del caso de estudio 2.

Requisitos	Requisito Satisface 100%	Requisitos No Satisfacen 100%
R1. Registrar Productos	√	
R2. Registrar Cliente	√	
R3. Registrar Venta	√	
R4. Registrar Compras	√	
R5. Generar Comprobante	√	
R6. Aperturar Caja	√	
R7. Consultar Productos	√	
R8. Consultar Ventas	√	
R9. Consultar Stock	√	
R10.Consultar Kárdex		√
Total:	9	1

Métrica: A/B = 9/10 = 0.9
Donde:
A = Nro. de requisitos satisfacen 100% las necesidades del usuario
B = Nro. total de requisitos

En la Tabla 6.40 se puede observar que el resultado de la métrica es 0.9, esto quiere decir que los requisitos satisfacen en un 90% las necesidades del usuario.

- **Métrica de calidad: Reusabilidad**

Aplicando la métrica de calidad MC003, explicado en la sección 6.2.3, se obtiene el resultado que se muestra en la Tabla 6.41.

Tabla 6.41. Resultados obtenidos al aplicar la métrica de calidad MC003 a los requisitos obtenidos aplicando Win Win del caso de estudio 2.

Requisitos	Requisitos Reusados	Requisitos no Reusados
R1. Registrar Productos	√	
R2. Registrar Cliente	√	

R3. Registrar Venta	√	
R4. Registrar Compras		√
R5. Generar Comprobante		√
R6. Aperturar Caja		√
R7. Consultar Productos		√
R8. Consultar Ventas		√
R9. Consultar Stock		√
R10.Consultar Kárdex		√
Total:	3	7

Métrica: A/B = 2/10 = 0.3
Donde:
A = Nro. de requisitos reusados para identificar otros requisitos
B = Nro. total de requisitos

En la Tabla 6.41 se puede observar que el resultado de la métrica es 0.3, esto quiere decir que los requisitos son reusados en un 30 % para capturar requisitos en nuevos proyectos.

- **Métrica de calidad: Eficiencia**

En la Tabla 6.42 se muestra el tiempo invertido en la identificación de los requisitos del caso de estudio 2, aplicando la técnica Win Win.

Tabla 6.42. Tiempo invertido en la captación de requisitos aplicando Win Win del caso de estudio 2.

Requisitos	Tiempo en minutos
Registrar productos	105
Registrar Cliente	90
Registrar Venta	115
Registrar Compras	100
Generar Comprobante	60
Aperturar Caja	30
Consultar Productos	45
Consultar Ventas	40

Consultar Stock	45
Consultar Kárdex	45
Totales	**675**

6.5.3 Resumen de Resultados

En la Tabla 6.43 se muestra el tiempo invertido en realizar la tarea de Captación de Requisitos aplicando *Casos de Usos, Win Win* y el *Modelo CBR* propuesto en los dos casos de estudio.

Según los resultados de la Tabla 6.43, se puede observar que el *Modelo CBR* propuesto aplicado en el caso de estudio 1, reduce en 75.10% el tiempo de captación de requisito obtenido por la técnica *Casos de Uso* y en 69.76 % el tiempo de captación de requisito obtenido por la técnica *Win Win.*

En la Tabla 6.43 también se puede observar que el *Modelo CBR* propuesto aplicado en el caso de estudio 2, reduce en 71.08% el tiempo de captación de requisito obtenido por la técnica *Casos de Uso* y en 63.41 % el tiempo de captación de requisito obtenido por la técnica *Win Win.*

Tabla 6.43. Cuadro comparativo del tiempo invertido en la captación de requisitos usando *Casos de Usos, Win Win* y el *modelo CBR propuesto* en ambos casos de estudio.

Caso de estudio 1				**Caso de estudio 2**			
Requisitos	**Caso de Uso**	**Win Win**	**Modelo CBR**	**Requisitos**	**Caso de Uso**	**Win Win**	**Modelo CBR**
Registrar Empleado	105	90	26	Registrar Productos	125	105	23
Aperturar Planilla	45	35	15	Registrar Cliente	110	90	23
Cerrar Planilla	45	40	15	Registrar Venta	130	115	35
Generar Planilla	180	150	43	Registrar Compras	120	100	35
Generar Boleta	80	60	27	Generar Comprobante	80	60	25
Generar Liquidaciones	120	100	42	Aperturar Caja	45	30	14
Registrar Licencias	110	90	24	Consultar Productos	60	45	16

Registrar Vacaciones	120	100	22	Consultar Ventas	50	40	14
Imprimir Planilla	95	75	15	Consultar Stock	60	45	19
Generar PDT	120	100	25	Consultar Kárdex	65	45	20
Totales	1020	840	254		845	675	224
(*) Tiempo en minutos							

La Tabla 6.44 muestra los resultados al aplicar las Métricas de Calidad a los requisitos obtenidos con el modelo CBR propuesto, Casos de Usos y Win Win en los dos casos de estudio.

Como se puede observar en la Tabla 6.44, los requisitos obtenidos en el caso de estudio 1 y aplicados el modelo CBR propuesto son entendibles en un 100%, la funcionalidad de estos requisitos satisfacen a los usuarios en un 100 % y estos requisitos son reusados en un 80%.

También se puede observar en esta tabla que los requisitos obtenidos en el caso de estudio 2 y aplicados el modelo CBR propuesto son entendibles en un 100%, la funcionalidad de estos requisitos satisfacen a los usuarios en un 100 % y estos requisitos son reusados en un 80%.

Tabla 6.44. Cuadro comparativo de los requisitos obtenidos con el modelo CBR propuesto, Casos de Usos y Win Win aplicando las métricas de calidad en ambos casos de estudio

Modelo CBR propuesto			
Código	**Nombre de Métrica de Calidad**	**Requisitos obtenidos en el caso de estudio 1**	**Requisitos obtenidos en el caso de estudio 2**
MC001	Entendibilidad	100%	100%
MC002	Completitud Funcional	100%	100%
MC003	Reusabilidad	80%	80%
Casos de Usos			
MC001	Entendibilidad	100%	100%
MC002	Completitud Funcional	90%	90%

MC003	Reusabilidad	10%	20%
Win Win			
MC001	Entendibilidad	100%	100%
MC002	Completitud Funcional	90%	90%
MC003	Reusabilidad	10%	30%
Lo más cercano a 100% es lo mejor			

En la Tabla 6.44 también se puede observar que los requisitos obtenidos en los dos casos de estudios, aplicando *Casos de Usos* y *Win Win*, son entendibles en un 100% y la funcionalidad de estos requisitos satisface a los usuarios en un 90%. Pero, estos requisitos se reusan con un porcentaje menor, en comparación con los requisitos obtenidos con el *modelo CBR propuesto*.

CAPÍTULO 7

CONCLUSIONES Y TRABAJOS FUTUROS

7.1 Conclusión General

El presente trabajo de investigación tiene como objetivo general proponer un modelo para la Captación de Requisito en el desarrollo de proyectos de software aplicando Razonamiento Basado en Casos (CBR).

Dicho objetivo se ha cumplido al presentar el modelo CBR expuesto ampliamente en el capítulo 4, en el cual se pudo observar el éxito del empleo de esta propuesta, analizando y comparando los resultados obtenidos en los dos casos de estudio y aplicando también otras dos técnicas: Casos de Usos y Win Win, y métricas de calidad expuestas en el capítulo 6.

7.2 Conclusiones Específicas

7.2.1 Objetivo Específico 1

Para el cumplimiento del objetivo específico 1, se analizaron las técnicas existentes para la Captación de Requisitos. Dicho estudio se realizó en el capítulo 3, donde se estudiaron las siguientes técnicas: *Entrevistas [Andriano 2006], Encuestas [Green 2003], Brainstorming [Raghavan+ 1994], Casos de Uso [Booch+ 1999], JAD [Dube+ 2010], Prototipacion [Andriano 2006] y Win Win [Chao+ 2009]*.

7.2.2 Objetivo Específico 2

Para el cumplimiento del objetivo específico 2, se estudiaron modelos de Razonamiento Basado en Casos existentes. Dicho estudio se realizó en el capítulo

2, donde se analizaron: el *Modelo de procesos de Ciclo de vida CBR*, propuesto por [Aamodt 1994], y el *Modelo de Descomposición de Tareas-Métodos*, también propuesto por [Aamodt 1994].

7.2.3 Objetivo Específico 3

Para el cumplimiento del objetivo específico 3, se escogió el modelo de CBR más adecuado para nuestra propuesta. Se escogió el *Modelo del Ciclo de vida CBR*, propuesto por [Aamodt 1994], como base para el desarrollo de nuestra propuesta. Se escogió este modelo debido a que está compuesto por cuatro pasos: engloba un ciclo de razonamiento continuo, busca en Librería de Casos problemas similares, adapta sus soluciones al problema y, finalmente, retroalirnenta la Librería de Casos con el nuevo problema resuelto. También se escogió el *Modelo de D*escomposición Tareas-Métodos propuesto por [Aamodt 1994], donde se definieron las tareas y métodos más adecuados a nuestro modelo.

7.2.4 Objetivo Específico 4

Para el cumplimiento del objetivo específico 4, se estudiaron las herramientas existentes en el mercado para el desarrollo de sistemas de Razonamiento Basado en Casos. Dicho estudio se realizó en el capítulo 2. Por ejemplo, se expuso las siguientes herramientas: *FreeCBR [FreeCbr 2011], CBRshell [AIAI 2010], CBR*Tools [INRIA 2011], jColibrí [GAIA 2011]*, etc.

7.2.5 Objetivo Específico 5

Para el cumplimiento del objetivo específico 5, se analizaron tres trabajos donde se han aplicado CBR en la ingeniería de Software. Dicho análisis se realizó en el capítulo 3. Los trabajos son los siguientes: *Uso de CBR en la Optimización de Calidad del Software*, propuesto por [Brady+ 2010], *Uso de CBR en el Diseño de Controles para Sistemas de Información basados en Internet (IIS)*, propuesto por [Lee+ 2009], y *Uso de CBR para la Estimación del Esfuerzo en el desarrollo de aplicaciones Web Hipermedia*, propuesto por [Mendes+ 2002].

7.2.6 Objetivo Específico 6

Para el cumplimiento del objetivo específico 6, se desarrolló una herramienta para poder implementar el modelo propuesto. Dicha herramienta se expuso en el capítulo 5, donde se explicó cómo se construyó la herramienta y cuáles son los pasos a seguir para configurar y construir nuestro modelo propuesto.

7.2.7 Objetivo Específico 7

Para el cumplimiento del objetivo específico 7, se desarrollaron dos casos de estudios: el *Sistema Integrado de Recurso Humanos (SIRH)* y el *Sistema de Punto de Venta (SPTA).* En ambos casos de estudios se realizó la Captación de diez requisitos usando: *Casos de Uso, Win Win* y el *Modelo CBR* propuesto.

7.2.8 Objetivo Específico 8

Para el cumplimiento del objetivo específico 8, se aplicaron métricas de Calidad a los requisitos obtenidos con *Caos de Usos, Win Win* y el *Modelo CBR propuesto*. Donde se observó que el *Modelo CBR* propuesto aplicado en los dos caso de estudio reduce en 73.09% el tiempo de captación de Requisito obtenido por la técnica *Casos de Uso* y en 66.58 % el tiempo de Captación de Requisito obtenido por la técnica *Win Win.* También se observó que los requisitos obtenidos con el *Modelo CBR* propuesto son entendibles en 100%, la funcionalidad de estos requisitos satisfacen a los usuarios en 100 %. También se ha podido observar que estos requisitos son reusados en 80%, mientras que con *Casos de Usos* en 15% y con *Win Win* en 20%, según los resultados obtenidos (ver Tabla 6.44).

7.3 Otras Conclusiones

Los estudios muestran que una de las causas de los atrasos de los proyectos de software se encuentra en la *Captación de Requisitos*. Por ello existen muchos esfuerzos para mitigar este problema, tales como métodos y técnicas para la captación de requisitos. Sin embargo, a pesar de estos esfuerzos, según los últimos reportes siguen los problemas.

CBR se ha aplicado con éxito en varios dominios, tales como: en la planificación, en los diagnósticos, en el diseño, etc. También se ha podido observar el éxito de esta técnica en la Ingeniería de Software.

Al Aplicar CBR en la Captación de Requisitos en el desarrollo de proyectos de software, se pudo observar que se aprovecharon los requisitos funcionales de proyectos de software desarrollados anteriormente, para luego resolver o identificar requisitos funcionales de nuevos proyectos.

7.4 Trabajos Futuros

Los trabajos futuros que podemos considerar son los siguientes:

- Integrar este modelo con las otras tareas de ingeniería de Requisitos, puesto que esta propuesta se centra solamente en la tarea *Captación*. Es decir, integrar este modelo con las tareas de Análisis, Especificación y Validación.

- Agregar pesos a los casos que han dado solución a problemas anteriores, en la etapa de Recuperación del ciclo de vida CBR. Es decir, los requisitos que den solución a un nuevo problema, ponderarlos, para así optimizar más la búsqueda.

- Cuando se representa el requisito se asignan pesos, estos pesos son asignados de acuerdo al criterio y experiencia del analista. Ello puede conllevar a errores. Por eso se recomienda usar *Redes Neuronales Artificiales* para la asignación de estos pesos. En el artículo [Perry+ 20000] se ve el éxito del uso de esta técnica, donde los autores usan esta técnica en la asignación de los pesos para la restauración de imágenes digitales.

- En la etapa *Retener*, el analista, de acuerdo a los cambios que se ha realizado a la solución del requisito, toma su criterio para retenerlos en la LR. Se recomienda usar alguna técnica, que ayude al analista ha tomar esta decisión, de esta forma, estaremos evitando llenar la LR con requisitos repetidos y optimizamos nuestra LR.

Referencias Bibliográficas

[Aamodt+ 1994] Aamodt A. and Plaza E., "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches", Artificial Intelligence Communications, Vol. 7, (1994), pp.39-52.

[Andriano 2006] Andriano N., "Comparación del Proceso de Elicitación de Requerimientos en el desarrollo de Software a Medida y Empaquetado. Propuesta de métricas para la Elicitación". Tesis Magistral, Universidad Blas Pascal Córdoba, Argentina, (2006).

[AIAI 2010] Artificial Intelligence Applications Institute, "CBR Shell", (10/12/2010), http://www.aiai.ed.ac.uk/technology/casebasedreasoning.html

[AliceSoft 2010] AliceSoft, "ReCall", (10/12/2010), http://www.alice-soft.com/html/prod_recall.htm

[Althoff 2001] Althoff Klaus-Dieter. Case-Based Reasoning. Handbook of Software Engineering and Knowledge Engineering. Institute for Experimental Software Engineering (IESE). (2001). http://www.iis.uni-hildesheim.de/files/staff/althoff/Publications/althoff-CBR-update.pdf

[Bareiss 1998] Bareiss, R., "PROTOS; a unified approach to concept representation, classification and learning", Univerity of Texas, Tesis doctoral, (1988), Texas - USA.

[Boehm + 1979] Boehm, B., Brown J., Kasper H., Lipow M., MacLeod G. and Merritt M., "Characteristics of Software Quality", IEEE Transactions on Software Engineering, Vol. 5, (1979), pp. 96-103.

[Boehm+ 1998] Boehm B., Egyed A., Kwan J., Port D., Shah A. and Madachy R., "Using the Win Win Spiral Model: A Case Study", IEEE Computer, pp. 33-44, (1998). http://sunset.usc.edu/aegyed/publications/-Using_the_WinWin_Spiral_Model-A_Case_Study.pdf

[Booch+ 1999] Booch G., Rumbaugh J. and Jacobson I., "The Unified Modeling Language User Guide", Addison–Wesley, (1999).

[Brady+ 2010] Brady A. and Menzies T., "Case-Based Reasoning vs Parametric Models for Software Quality Optimization", ACM, (2010), pp. 1-10.

[Borland 2011] Borland, "CALIBER-RM", (05/03/2011), http://www.borland.com/us/products/caliber/

[Chao+ 2009] Chao D. and Ming W., “Distributed Requirement Elicitation and Negotiation Based on the Hall for Workshop of Meta-Synthetic Engineering”, ACM, (2009), pp. 1-5. http://dl.acm.org/citation.cfm?id=1656193

[Christel+1992] Christel M. and Kang K., “Issues in Requirements Elicitation”, Technical Report, Software Engineering Institute, Pennsylvania, (1992).

[Dijkman+ 2008] Dijkman R., Dumas M. and Ouyang CH., “Semantics and analysis of business process models in BPMN”, ACM, Volume 50 Issue 12, (2008).

[Dube+ 2010] Dube R. and Dixit S., “Process-oriented Complete Requirement Engineering Cycle for Generic Projects”, International Conference and Workshop on Emerging Trends in Technology, ACM, India, (2010).

[Durán 2011] Durán M., “REM - Requisite Management”, (05/04/2011), http://www.lsi.us.es/descargas/descarga_programas.php?id=3

[Durán+ 2000] Durán A. y Bernárdez B., “Metodología para la Elicitación de Requisitos de Sistemas Software”, Informe Técnico LSI–2000–10, Sevilla, España. (2000).

[FreeCBR 2011] FreeCbr, (15/03/2011), http://freecbr.sourceforge.net/

[GAIA 2011] Group for Artificial Intelligence Applications, “jColibrí”, (5/01/2011), http://gaia.fdi.ucm.es/grupo/projects/jcolibri/

[Green 2003] Stewart Green S., “Eliciting Stakeholders s’ Knowledge of Goals and Processes to derive IT Support”, The University of the West of England (2003).

[Goel 1989] Goel, A., “Integración of case-based reasoning and rnodel-based reasoning for adaptation”, Ohio Univeristy, Ohio University, Tesis doctoral, (1989), Ohio - USA.

[Grillo+ 2009] Grillo O. y La Rosa M., “Sistema Administrador de Requerimientos y Planificador de Tareas”, Pontificia Universidad Católica del Perú, Tesis, (2009), Lima – Perú.

[Hammond 1986] Hammond, K., “CHEF: a model of case-based planning”, Proceedings of the Fourth National Conference on Artificial Intelligence, (1986).

[Hinrichs+ 1991] Hinrichs T. and Kolodner J., “The roles of adaptation in case-based design”, Proceedings of the Ninth National Conference on Articial Intelligence, Anaheim, (1991).

[Hughes 1997] Hughes, R. T., “An Empirical investigation into the estimation of software development effort”, PhD thesis, Dept. of Computing, the University of Brighton, (1997), Brighton - Inglaterra.

[IBM 2011] IBM, “Calculating your return on investment from more effective requirements management”, (02/01/2011), http://www-128.ibm.com/developerworks/rational/library/347.html

[IBM-DOORS 2011] IBM, “DOORS”, (04/05/2011), http://www-01.ibm.com/software/awdtools/doors/productline/

[IBM-RRC 2012] IBM, “Rational Requirements Composer”, (20/06/2012), http://www-142.ibm.com/software/products/es/es/rrc/

[INRIA 2011] INRIA, “CBR*Tools”, (23/01/2011), http://www-sop.inria.fr/axis/cbrtools/

[IEEE 1990] IEEE, “Standard Glossary of Software Engineering Terminology”, Computer Society of IEEE, (1990).

[ISO 2011] Norma ISO/IEC TR 9126-3: 2003 - Software engineering - Product quality - Part 3: Internal metrics, (2011).

[Jacobson+ 1993] Jacobson I., Christerson M., Jonsson P. and Overgaard G., “Object–Oriented Software Engineering: A Use Case Driven Approach”, Addison–Wesley, 4a edición, (1993).

[Juárez+ 2005] Juárez J. y Palma J., “Inteligencia Artificial”: Razonamiento Basado en Casos, Universidad de Murcia, Vol. 74, (2005), Murcia-España.

[Koton 1989] Koton, P., “Using experience in learning and problem solving”, Informe técnico AÍIT/LCS/TR-441, MIT, (1989).

[Kotonya + 2001] Kotonya G., Sawyer P., “Software Requirements”, IEEE – Trial Version 1.0 (2001), pp. 1-26.

[Lasheras+ 2011] Lasheras J., Álvarez J., Nicolás J., Moros B., “Soporte Automatizado a la Reutilización de Requisitos”, IEEE, (2003), pp. 335-346. http://www.um.es/giisw/docs/SirenTool.pdf

[Lee+ 2009] Lee S. and Kim K., “Using case-based reasoning for the design of controls for internet-based information systems”, Sciencedirect, (2009), Volume 36, Issue 3, Part 1. Pages 5582–5591.
http://www.sciencedirect.com/science/article/pii/S0957417408004119

[Leite 1987] Leite, J., “A Survey on Requirements Analysis”, Advancing Software Engineering Project Technical Report RTP-071, University of California at Irvine, Department of Information and Computer Science, (1987).

[López+ 1993] López, B. and Plaza, E., “Case-based learning of plans and goal states in medical diagnosis”, Artificial Intelligence in Medicine, Vol. 9, (1993), pp. 29-60.

[Mendes+ 2000] Mendes, E., Counsell, S. and Mosley, N., “Measurement and Effort Prediction of Web Applications”, Proc. Second ICSE Workshop on Web Engineering, (2000).

[Mendes+ 2001] Mendes, E., Counsell, S., and Mosley, N., “Towards the Prediction of Development Effort for Hypermedia Applications”, Proceedings of the 12th ACM conference on Hypertext and Hypermedia, (2001).

[Mendes+ 2002] Mendes E., Mosley N. and Watson I. “A comparison of case-based reasoning approaches to Web hypermedia Project Cost Estimation”, ACM, (2002), pp. 1-9. http://portal.acm.org/citation.cfm?id=511446.511482&coll=DL&dl=GUIDE&CFID=31271263&CFTOKEN=12937670

[Mindbox 2010] Mindbox, “Art*Eiiterprise”, (10/12/2010), http://www.mindbox.com/

[Oberg+ 2000] Oberg R., Probasco L. and Ericsson M., “Applying Requirements Management with Use Cases”, Rational Software Corporation, Technical Paper TP505, (2000).

[Perry+ 2000] Perry, S. W. and Guan, L., “Weight assignment for adaptive image restoration by neural networks”, IEEE Computational Intelligence Society, Vol. 11, (2000), pp. 156 – 170.

[Piattini+ 1996] Piattini M., Calvo J., Cervera J. y Fernández L., “Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión”, Madrid, España, (1996).

[Raghavan et al. 1994] S. Raghavan, G. Zelesnik, and G. Ford. Lecture Notes on Requirements Elicitation. Educational Materials CMU/SEI–94–EM–10, Software Engineering Institute, Carnegie Mellon University, 1994, http://www.sei.cmu.edu.

[Rational 1998] Rational, “Rational Unified Process Best Practices for Software Development Teams”, IBM (Rational Software Company), (1998), http://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf

[Rational 2011] Rational RequsitePro, “RequsitePro”, (02/03/2011), http://www-01.ibm.com/software/awdtools/reqpro/

[Rossillea+ 1993] Rossillea D., Laurentc J. y Burguna A., “Modelling a Decision-Support System for Oncology using Rule-Based and Case-Based

Reasoning Methodologies", International Journal of Medical Informatics, Vol. 74, (2005), pp. 2-4.

[Schank 1982] Schank, R., "Dynamic Memory", Cambridge Unviersity Press, (1982), Cambridge - Inglaterra.

[Schulz 1999] Schulz, S., "CBR-Works – A State-of-the-Art Shell for Case- Based Application Building", Proceedings of the German Workshop on Case-Based Reasoning, (1999).

[Sheldon+ 1992] Sheldon, F., Kavi K., Tausworth R., Brettschneider R. and Everett W., "Reliability Measurement from Theory to Practice", IEEE Software, Vol. 9, (1992).

[Shiu+ 2004] Shui S, Pal S., "Foundations of Soft Case-based reasoning", Wiley Series On Intelligent Systems, Hoboken, New Jersey (2004).

[StandishGroup 1995] Standish Group, "The Chaos Report 1995 - 2004", (10/10/2011), https://cs.nmt.edu/~cs328/reading/Standish.pdf

[StandishGroup 2009] Standish Group, "The Chaos Report 2009", (10/10/2011), http://www1.standishgroup.com/newsroom/chaos_2009.php

[Stottler 2011] Stottler Henke Associates, "ESTEEM", (20/02/2011), http://www.stottlerhenke.com/solutions/decision_support/esteem.htm.

[Sycara 1992] Sycara, K.: "CADET: a case-based syntehsis tool for engineering design", International Journal for Expert Systems, Vol. 4(ii), (1992), pp. 157-188.

ANEXOS

Anexo 1: Carga de Requisitos de ocho Proyectos de Software a la *Librería de Requisitos*

Requisitos

Proyectos

Nombre | Empresa | Estado del proyecto

Requisitos

ID	Proyecto	Dominio	Modulo	Verbo	Sustantivo	Actor 1	Actor 2	Area	SR	Estado	Caso ID Sol.
1	SISTEMA DE VENTAS	COMERCIAL	VENTAS	REGISTRAR	CLIENTE	VENDEDO	ADMINIST	VENTAS	SRP6-1	CARGADO	
2	SISTEMA DE PLANILLAS	COMERCIAL	PLANILLAS	REGISTRAR	EMPLEADO	ASISTENTE	ENCARGADO	RECURSOS	SRP3-1	CARGADO	
3	SISTEMA DE MATRICULA	EDUCACION	MATRICULA	REGISTRAR	ALUMNO	SECRETARIA	ALUMNOS	POSTGRADO	SRP7-1	CARGADO	
5	SISTEMA DE CONTROL DE INVENTARIO	COMERCIAL	INVENTARIO	REGISTRAR	INVENTARI	ENCARGADO	ASISTENTE	ALMACEN	SRP2-1	CARGADO	
6	SISTEMA DE PLANILLAS	COMERCIAL	PLANILLAS	APERTURAR	PLANILLA	ASISTENTE	ENCARGADO	RECURSOS	SRP3-2	CARGADO	
7	SISTEMA DE PLANILLAS	COMERCIAL	PLANILLAS	CERRAR	PLANILLA	ASISTENTE	ENCARGADO	RECURSOS	SRP3-3	CARGADO	
8	SISTEMA DE PLANILLAS	COMERCIAL	PLANILLAS	GENERAR	PLANILLA	ASISTENTE	ENCARGADO	RECURSOS	SRP3-4	CARGADO	
9	SISTEMA DE PLANILLAS	COMERCIAL	PLANILLAS	GENERAR	BOLETA	ASISTENTE	ENCARGADO	RECURSOS	SRP3-5	CARGADO	
10	SISTEMA DE PLANILLAS	COMERCIAL	PLANILLAS	GENERAR	LIQUIDACIO	ASISTENTE	ENCARGADO	RECURSOS	SRP3-6	CARGADO	
11	SISTEMA DE PLANILLAS	COMERCIAL	PLANILLAS	REGISTRAR	LICENCIAS	ASISTENTE	ENCARGADO	RECURSOS	SRP3-7	CARGADO	
12	SISTEMA DE PLANILLAS	COMERCIAL	PLANILLAS	REGISTRAR	VACACIONE	ASISTENTE	ENCARGADO	RECURSOS	SRP3-8	CARGADO	
13	SISTEMA DE PLANILLAS	COMERCIAL	PLANILLAS	IMPRIMIR	PLANILLA	ASISTENTE	ENCARGADO	RECURSOS	SRP3-9	CARGADO	
14	SISTEMA DE PLANILLAS	COMERCIAL	PLANILLAS	GENERAR	PDT	ASISTENTE	ENCARGADO	RECURSOS	SRP3-10	CARGADO	
25	SISTEMA DE PLANILLAS	COMERCIAL	PLANILLAS	REGISTRAR	CONTRATO	ASISTENTE	ENCARGADO	RECURSOS	SRP3-11	CARGADO	
15	SISTEMA DE VENTAS	COMERCIAL	VENTAS	REGISTRAR	VENTA	VENDEDOR	ENCARGADO	VENTAS	SRP6-2	CARGADO	
16	SISTEMA DE VENTAS	COMERCIAL	VENTAS	GENERAR	COMPROBA	VENDEDOR	ENCARGADO	VENTAS	SRP6-3	CARGADO	

Requisitos cargados a la LR de ocho proyectos de software.

Requisitos

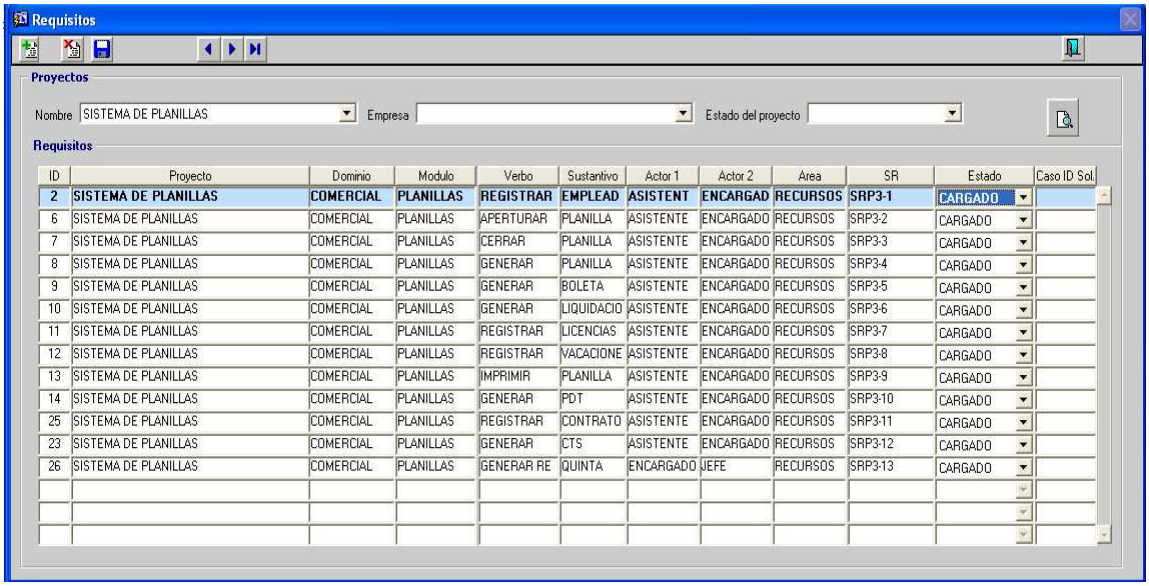
Proyectos

Nombre SISTEMA DE VENTAS | Empresa | Estado del proyecto

Requisitos

ID	Proyecto	Dominio	Modulo	Verbo	Sustantivo	Actor 1	Actor 2	Area	SR	Estado	Caso ID Sol.
1	SISTEMA DE VENTAS	COMERCIAL	VENTAS	REGISTRAR	CLIENTE	VENDEDO	ADMINIST	VENTAS	SRP6-1	CARGADO	
15	SISTEMA DE VENTAS	COMERCIAL	VENTAS	REGISTRAR	VENTA	VENDEDOR	ENCARGADO	VENTAS	SRP6-2	CARGADO	
16	SISTEMA DE VENTAS	COMERCIAL	VENTAS	GENERAR	COMPROBA	VENDEDOR	ENCARGADO	VENTAS	SRP6-3	CARGADO	
17	SISTEMA DE VENTAS	COMERCIAL	VENTAS	APERTURAR	CAJA	VENDEDOR	ENCARGADO	VENTAS	SRP6-4	CARGADO	
18	SISTEMA DE VENTAS	COMERCIAL	VENTAS	CERRAR	CAJA	VENDEDOR	ENCARGADO	VENTAS	SRP6-5	CARGADO	
19	SISTEMA DE VENTAS	COMERCIAL	VENTAS	CONSULTAR	VENTAS	VENDEDOR	ADMINISTRA	VENTAS	SRP6-6	CARGADO	
20	SISTEMA DE VENTAS	COMERCIAL	VENTAS	CONSULTAR	CLIENTES	VENDEDOR	ENCARGADO	VENTAS	SRP6-7	CARGADO	
21	SISTEMA DE VENTAS	COMERCIAL	VENTAS	CONSULTAR	EMPLEADO	VENDEDOR	ADMINISTRA	VENTAS	SRP6-8	CARGADO	
22	SISTEMA DE VENTAS	COMERCIAL	VENTAS	IMPRIMIR	COMPROBA	VENDEDOR	ENCARGADO	VENTAS	SRP6-9	CARGADO	

Requisito del proyecto de software *Sistema de Ventas* cargados a la LR.



Requisitos

Proyectos

Nombre SISTEMA DE PLANILLAS Empresa Estado del proyecto

Requisitos

ID	Proyecto	Dominio	Modulo	Verbo	Sustantivo	Actor 1	Actor 2	Area	SR	Estado	Caso ID Sol.
2	SISTEMA DE PLANILLAS	COMERCIAL	PLANILLAS	REGISTRAR	EMPLEAD	ASISTENT	ENCARGAD	RECURSOS	SRP3-1	CARGADO	
6	SISTEMA DE PLANILLAS	COMERCIAL	PLANILLAS	APERTURAR	PLANILLA	ASISTENTE	ENCARGADO	RECURSOS	SRP3-2	CARGADO	
7	SISTEMA DE PLANILLAS	COMERCIAL	PLANILLAS	CERRAR	PLANILLA	ASISTENTE	ENCARGADO	RECURSOS	SRP3-3	CARGADO	
8	SISTEMA DE PLANILLAS	COMERCIAL	PLANILLAS	GENERAR	PLANILLA	ASISTENTE	ENCARGADO	RECURSOS	SRP3-4	CARGADO	
9	SISTEMA DE PLANILLAS	COMERCIAL	PLANILLAS	GENERAR	BOLETA	ASISTENTE	ENCARGADO	RECURSOS	SRP3-5	CARGADO	
10	SISTEMA DE PLANILLAS	COMERCIAL	PLANILLAS	GENERAR	LIQUIDACIO	ASISTENTE	ENCARGADO	RECURSOS	SRP3-6	CARGADO	
11	SISTEMA DE PLANILLAS	COMERCIAL	PLANILLAS	REGISTRAR	LICENCIAS	ASISTENTE	ENCARGADO	RECURSOS	SRP3-7	CARGADO	
12	SISTEMA DE PLANILLAS	COMERCIAL	PLANILLAS	REGISTRAR	VACACIONE	ASISTENTE	ENCARGADO	RECURSOS	SRP3-8	CARGADO	
13	SISTEMA DE PLANILLAS	COMERCIAL	PLANILLAS	IMPRIMIR	PLANILLA	ASISTENTE	ENCARGADO	RECURSOS	SRP3-9	CARGADO	
14	SISTEMA DE PLANILLAS	COMERCIAL	PLANILLAS	GENERAR	PDT	ASISTENTE	ENCARGADO	RECURSOS	SRP3-10	CARGADO	
25	SISTEMA DE PLANILLAS	COMERCIAL	PLANILLAS	REGISTRAR	CONTRATO	ASISTENTE	ENCARGADO	RECURSOS	SRP3-11	CARGADO	
23	SISTEMA DE PLANILLAS	COMERCIAL	PLANILLAS	GENERAR	CTS	ASISTENTE	ENCARGADO	RECURSOS	SRP3-12	CARGADO	
26	SISTEMA DE PLANILLAS	COMERCIAL	PLANILLAS	GENERAR RE	QUINTA	ENCARGADO	JEFE	RECURSOS	SRP3-13	CARGADO	

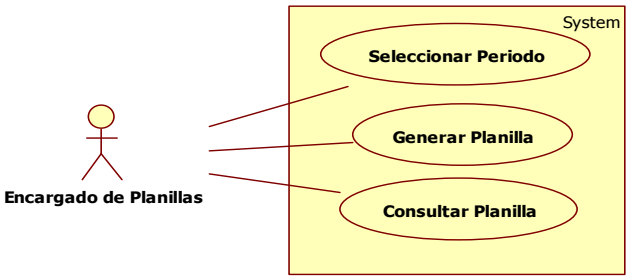
Requisito del proyecto de software *Sistema de Planillas* cargados a la LR.

Anexo 2: Captación de requisitos usando Casos de Uso del caso de estudio 1: Sistema Integrado de Recursos Humanos (SIRH)

Caso de uso *Generar Planilla* del caso de estudio 1: Sistema Integrado de Recursos Humanos (SIRH)

Caso de Uso: R2-Generar Planilla	**Analista:** Ediberto Iparraguirre

Diagrama de Casos de Uso



Breve Descripción

El sistema deberá permitir generar los diferentes tipos de Planilla de los empleados, tales como: Planilla Mensual, Planilla de Gratificaciones, Planilla de Vacaciones, etc. Para ello el periodo de la planilla debe estar aperturado. Para generar la planilla se debe calcular el total de ingresos, descuentos y aportaciones por cada empleado. También se debe permitir consultar la planilla.

Actores

- *Encargado de Planillas*: usuario interno a la organización que está encargado de calcular todos los tipos de las planillas de los trabajadores que laboran en la empresa.

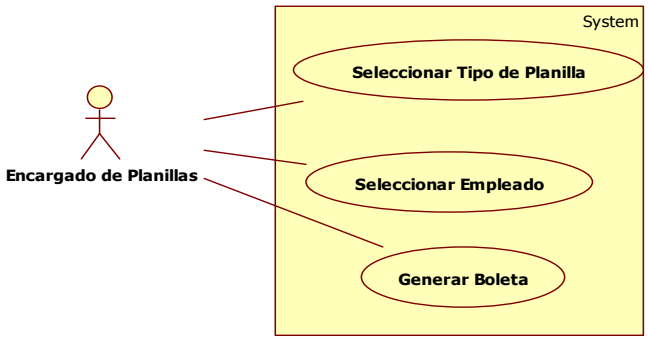
Flujo básico

- El usuario selecciona la opción "Generar Planilla". El sistema muestra el formulario respectivo.
- El usuario ingresa el periodo de la planilla: año y mes. El sistema verifica si el periodo ingresado es el correcto y si está abierto.
- Luego el usuario selecciona el tipo de planilla: Mensual o de Gratificaciones.
- Se activa el botón "Generar Planilla", el usuario hace clic en este botón. El sistema muestra un mensaje preguntándole si está seguro de generar la planilla. El usuario acepta, luego el sistema realiza todos los cálculos para la generación de la planilla de todos los empleados del periodo correspondiente.
- Luego el sistema muestra un mensaje, especificando que la planilla se generó con éxito.
- El usuario hace clic en la pestaña "Consultar Planilla", para verificar los datos de la planilla de todos los empleados.
- Finalmente, el usuario selecciona el botón "Salir" para cerrar el formulario.
- El caso de uso termina.

Caso de uso *Generar Boleta* del caso de estudio 1: Sistema Integrado de Recursos Humanos (SIRH)

Caso de Uso: R3-Generar Boleta	**Analista:** Ediberto Iparraguirre

Diagrama de Casos de Uso



Breve Descripción

El *Encargado de Planillas* habiendo terminado el procesamiento de la generación de alguna planilla, tiene que generar el reporte de la boleta de pago, relacionado a la respectiva planilla procesada. Para ello busca el tipo de planilla, el periodo y el empleado relacionado a la planilla.

Actores

- *Encargado de Planillas*: usuario interno a la organización que está encargado de calcular todos los tipos de las planillas de los trabajadores que laboran en la empresa.

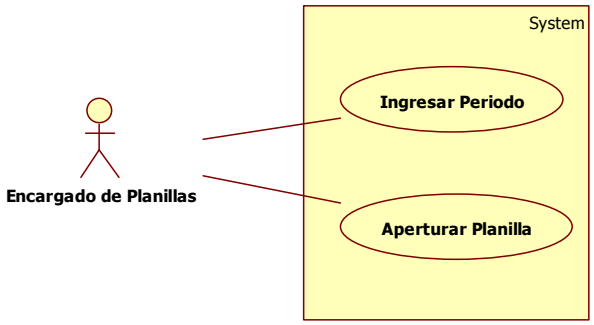
Flujo básico

- El usuario selecciona la opción "Generar Reporte Boleta de Pago".
- El sistema muestra el formulario respectivo con los parámetros de Periodo y Tipo de Planilla.
- El usuario selecciona el periodo de la planilla (año y mes) y Tipo de planilla (Planilla Mensual o Planilla de Gratificaciones).
- El sistema realiza la validación de si existe la planilla para el periodo indicado.
- En caso de que no exista la planilla indicada, el sistema muestra un mensaje de aviso en una ventana emergente, con el texto "Error: No se han encontrado la planilla para el periodo indicado". El usuario acepta el mensaje.
- Si existe la planilla, el sistema habilita el botón “Generar Reporte” y la lista, en donde se muestran los datos de los empleados que pertenecen a esa planilla en el periodo indicado.
- Por defecto se encuentra seleccionada la opción "Seleccionar:", el usuario puede seleccionar uno o más empleados. En caso de elegir la opción "Todos" el sistema marca el check automáticamente a todos los empleados de la lista para generar el reporte de boleta de pago.
- El usuario presiona el botón "Generar Reporte".
- Finalmente, el usuario presiona el botón "Salir" para cerrar el formulario.
- El caso de uso termina.

Caso de uso *Aperturar Planilla* del caso de estudio 1: Sistema Integrado de Recursos Humanos (SIRH)

Caso de Uso: R4-Aperturar planilla	**Analista:** Ediberto Iparraguirre

Diagrama de Casos de Uso



Breve Descripción

Esta funcionalidad deberá permitir abrir el periodo para luego poder realizar todas las tareas relacionadas al cálculo de la planilla, tales como: adelanto de quincena, adelanto de vacaciones, calcular porcentaje de sueldo para el pago de quincena, etc. Luego de aperturar la planilla el sistema queda en el estado habilitado para el registro de los ingresos y descuentos excepcionales, para la generación de la planilla y para todos los procesos asociados.

Actores

- *Encargado de Planillas*: usuario interno a la organización que está encargado de calcular todos los tipos de las planillas de los trabajadores que laboran en la empresa.

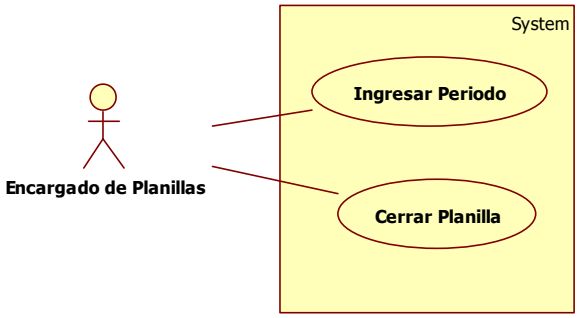
Flujo básico

- El usuario selecciona la opción "Generar Reporte Boleta de Pago".
- El usuario selecciona la opción “Aperturar Planilla”.
- El sistema muestra el formulario “Aperturar Planilla”.
- El usuario selecciona el tipo de planilla a aperturar: mensual, quincenal, gratificaciones y/o vacaciones.
- Finalmente, el usuario selecciona el botón “Ok” para ejecutar la apertura de la planilla.
- El caso de uso termina.

Caso de uso *Cerrar Planilla* del caso de estudio 1: Sistema Integrado de Recursos Humanos (SIRH)

Caso de Uso: R5-Cerrar planilla	**Analista:** Ediberto Iparraguirre

Diagrama de Casos de Uso



Breve Descripción

El *Encargado de Planillas* se encarga de ejecutar el cierre de planilla correspondiente a un periodo dado, dicho proceso sirve para indicar que se terminó de ejecutar todas las tareas relacionadas a la planilla en un determinado periodo. Luego del cierre de la planilla no se podrá realizar ningún cambio.

Actores

- *Encargado de Planillas*: usuario interno a la organización que está encargado de calcular todos los tipos de las planillas de los trabajadores que laboran en la empresa.

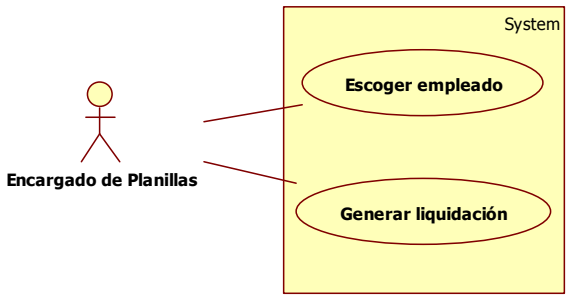
Flujo básico

- El usuario selecciona la opción "Cerrar Planilla". El sistema verifica internamente si se han realizado los procesos que son prerequisitos. Los procesos son los siguientes: Impresión de libro de planillas, Generación del PDT de planillas, Envío del PDT a la SUNAT, Verificación del cierre, Actualización del cierre y Cierre de planilla.
- Finalmente, el usuario selecciona el botón "Salir" para cerrar el formulario.
- El caso de uso termina.

Caso de uso *Generar Liquidaciones* del caso de estudio 1: Sistema Integrado de Recursos Humanos (SIRH)

Caso de Uso: R6-Generar liquidaciones	**Analista:** Ediberto Iparraguirre

Diagrama de Casos de Uso



Breve Descripción

El *Encargado de Planilla* se encargará de ejecutar la liquidación de los empleados, esta incluye el cálculo de CTS, Vacaciones devengadas, Gratificaciones devengadas, así como pagos extras por conceptos varios (Bonificaciones, Premios, etc.). Dependiendo del motivo de cese de contrato al trabajador, se le deberá calcular adicionalmente una indemnización.

Actores

- *Encargado de Planillas*: usuario interno a la organización que está encargado de calcular todos los tipos de las planillas de los trabajadores que laboran en la empresa.

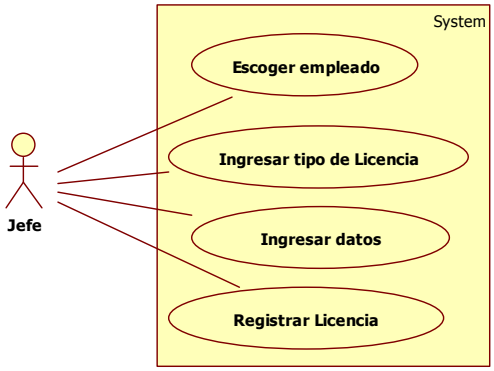
Flujo básico

- El usuario selecciona la opción del menú “liquidación de empleado”.
- El sistema solicita el ingreso del código de empleado que se desea liquidar.
- El usuario proporciona el código de empleado previa búsqueda.
- El sistema procesa la información del empleado y le solicita el ingreso de la fecha de cese, fecha de la liquidación (esta fecha se utiliza para el cálculo de intereses si el empleador no liquida al empleado el mismo día en que cesa sus funciones) y motivo de cese.
- El usuario ingresa la información solicitada fecha de cese, fecha de liquidación, motivo de cese y ejecuta la opción generar liquidación.
- El sistema realiza los siguientes procesamientos para generar la liquidación: Vacaciones Devengadas, Cálculo de CTS, Cálculo de Gratificaciones pendientes, Cálculo de Pagos por extras y la suma de los submontos obtenidos en cada proceso.
- Luego el sistema emite la hoja de liquidación por pantalla.
- Finalmente, el usuario selecciona el botón “Salir” para cerrar el formulario.
- El caso de uso termina.

Caso de uso *Registrar Licencia* del caso de estudio 1: Sistema Integrado de Recursos Humanos (SIRH)

Caso de Uso: R7-Registrar Licencias	**Analista:** Ediberto Iparraguirre

Diagrama de Casos de Uso



Breve Descripción

Este Caso de Uso se refiere a las actividades llevadas a cabo para completar satisfactoriamente el registro de las licencias de los trabajadores. El sistema deberá permitir registrar la programación de licencia por maternidad, enfermedad y sin goce de haber. Para ello se debe ingresar el tipo de licencia y los periodos (fecha de inicio y fecha de término).

Actores

- *Jefe del Personal*: es el usuario interno a la organización que está encargado de la administración del personal.

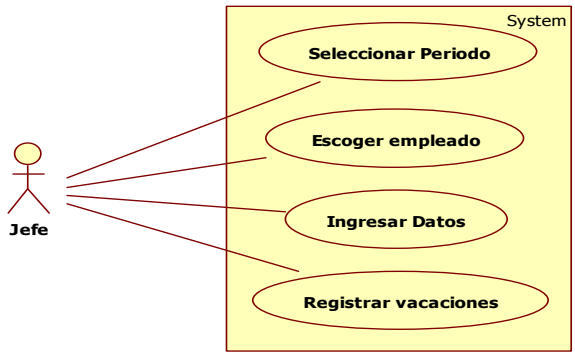
Flujo básico

- El usuario selecciona la opción Registrar la Licencia.
- El usuario procede a llenar los datos del trabajador al que desea ingresar la licencia, pudiendo ingresar su código o nombre del trabajador.
- El usuario procede con el llenado de los campos solicitados por el Sistema. La información que deberá incluir en este rubro son las siguientes: Tipo de Licencia (sin goce de haber, con goce de haber), Motivo (seleccionar de una lista el motivo de la salida), Fecha de emisión, Justificación, Rango de fecha inicial, Rango de fecha final, Número de días, Rango de hora inicial, Rango de hora final, número de horas trabajadas y número de horas no trabajadas.
- Finalmente, el usuario pulsa el botón Guardar y la papeleta de salida se registra satisfactoriamente y esta es tomada en cuenta para la generación de la tarea mensual.
- El caso de uso termina.

Caso de uso *Registrar Vacaciones* del caso de estudio 1: Sistema Integrado de Recursos Humanos (SIRH)

Caso de Uso: R8-Registrar Vacaciones	**Analista:** Ediberto Iparraguirre

Diagrama de Casos de Uso



Breve Descripción

Este caso de uso se refiere al registro de la programación de vacaciones de los empleados de la empresa, así como la consulta de los mismos, en un determinado periodo.

Actores

- *Jefe del Personal*: es el usuario interno a la organización que está encargado de la programación de vacaciones del personal.

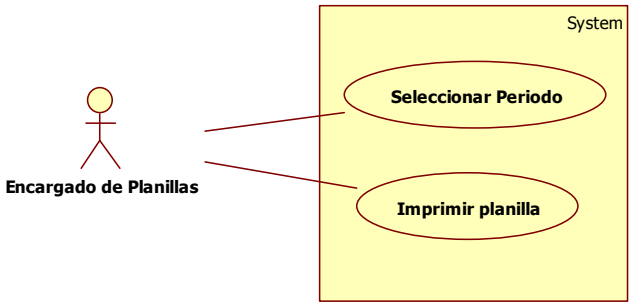
Flujo básico

- El usuario entra a la opción Asignar vacaciones, el usuario debe escoger el periodo de las vacaciones (año y mes).
- Luego el sistema mostrará una lista de los empleados que no tengan asignadas vacaciones en el periodo seleccionado.
- El usuario establecerá la fecha de inicio y fecha final de las vacaciones que serán asignado al empleado.
- El usuario marcara con check a los empleados que tendrán vacaciones en el periodo y fechas establecidas.
- El usuario, luego de seleccionar los empleados, hará click sobre el botón Registrar Vacaciones.
- El sistema registrará las vacaciones de los empleados en las fechas y periodo indicados.
- El caso de uso termina.

Caso de uso *Imprimir Planilla* del caso de estudio 1: Sistema Integrado de Recursos Humanos (SIRH)

Caso de Uso: R9-Imprimir Planilla	**Analista:** Ediberto Iparraguirre

Diagrama de Casos de Uso



Breve Descripción

Este caso de uso se refiere a las actividades llevadas a cabo para generar satisfactoriamente el reporte con la información de las planillas elaboradas en un determinado periodo, el cual es realizado por el Encargado de planillas.

Actores

Encargado de Planilla: es el usuario interno a la organización que está encargado de la programación de vacaciones del personal.

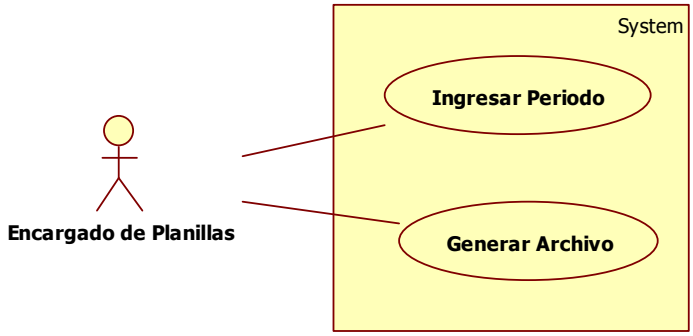
Flujo básico

- El usuario selecciona la opción "Imprimir Planilla". El sistema muestra el formulario respectivo con los filtros.
- El usuario selecciona el periodo de la planilla: año y mes. También selecciona el empleado o la opción Todos. El sistema verifica si el periodo ingresado es el correcto y si está cerrado.
- Se activa el botón "Generar Reporte", el usuario hace clic en este botón. El sistema muestra un mensaje preguntándole si está seguro de generar el reporte con los criterios seleccionados. El usuario acepta.
- Luego el sistema muestra el reporte con los criterios seleccionados: año y mes de la planilla, empleado o listado de empleados seleccionados, conceptos de la planilla y las cantidades monetarias correspondientes.
- Finalmente, el usuario selecciona el botón "Salir" para cerrar el reporte.
- El caso de uso termina.

Caso de uso *Generar PDT* del caso de estudio 1: Sistema Integrado de Recursos Humanos (SIRH)

Caso de Uso: R10-Generar PDT	**Analista:** Ediberto Iparraguirre

Diagrama de Casos de Uso



Breve Descripción

Este Caso de Uso se refiere a las actividades llevadas a cabo para generar el archivo con la información relevante según los requerimientos de la SUNAT. Este archivo se generará luego de terminar de realizar la planilla en un determinado periodo, para luego ser cargado en el PDT (Programa de Declaración Telemática) de la SUNAT. La generación de este archivo se hará en forma mensual después que se cierre el periodo de la planilla.

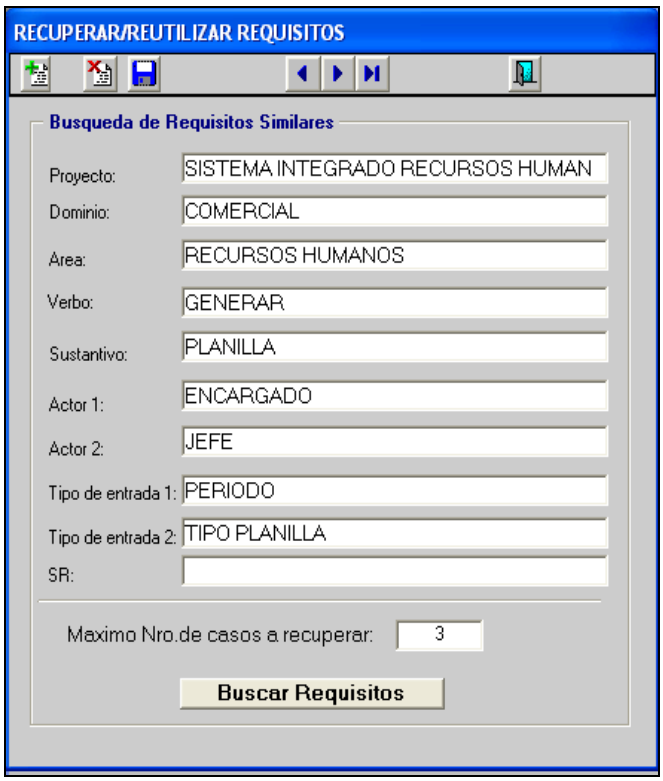
Actores

- *Encargado de Planilla*: es el usuario interno a la organización que está encargado de la programación de vacaciones del personal.

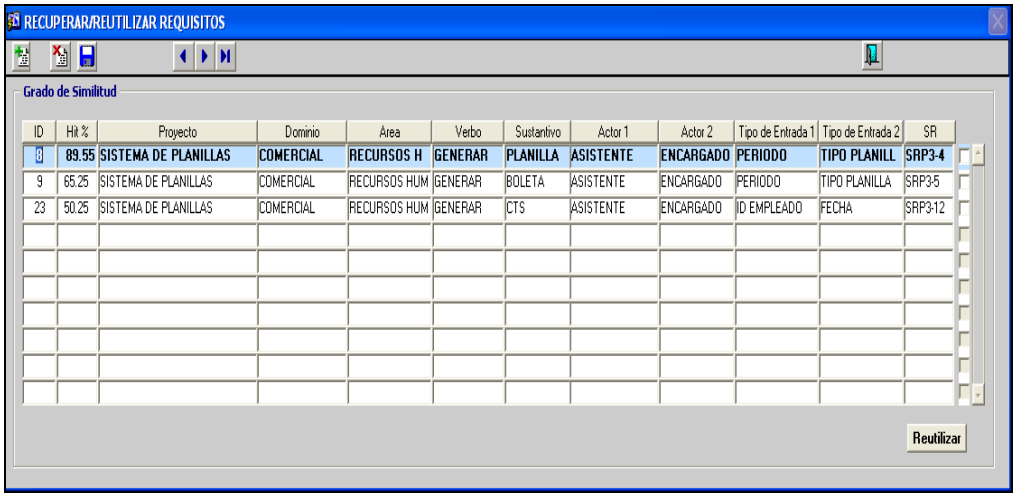
Flujo básico

- El usuario selecciona la opción "Generar PDT".
- El sistema muestra una pantalla pidiendo los siguientes datos: Tipo de planilla y periodo (año y mes).
- El usuario presiona el botón: "Generar archivo".
- Luego el sistema muestra un mensaje informando que el archivo se generó satisfactoriamente.
- Finalmente, el usuario selecciona el botón "Salir" para cerrar el reporte.
- El caso de uso termina.

Anexo 3: Captación de requisitos usando el modelo *CBR* propuesto del caso de estudio 1: Sistema Integrado de Recursos Humanos (SIRH)

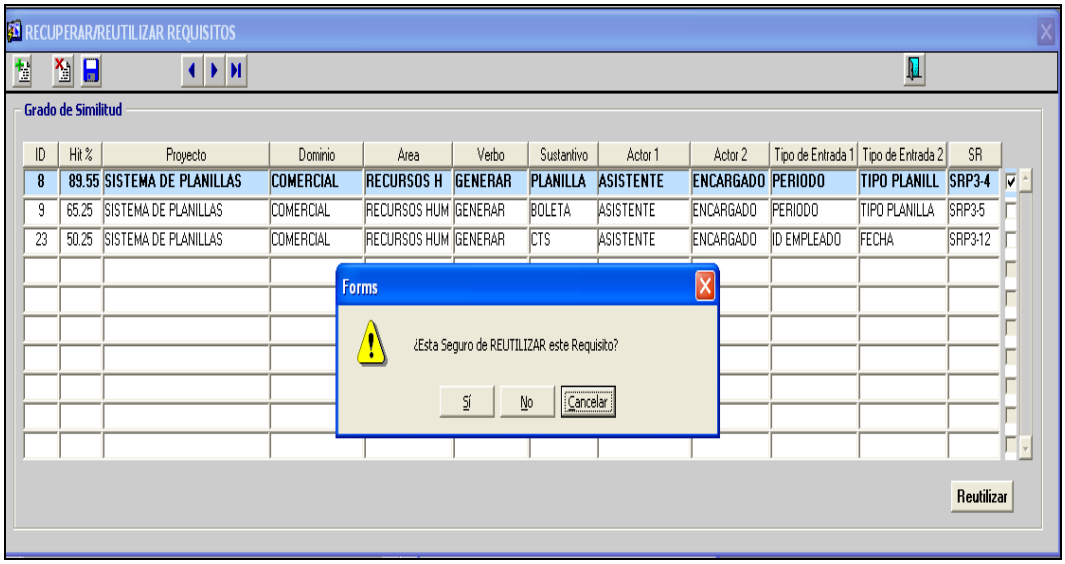


Recuperación del requisito *Generar Planilla* del caso de estudio 1, utilizando el modelo CBR propuesto.

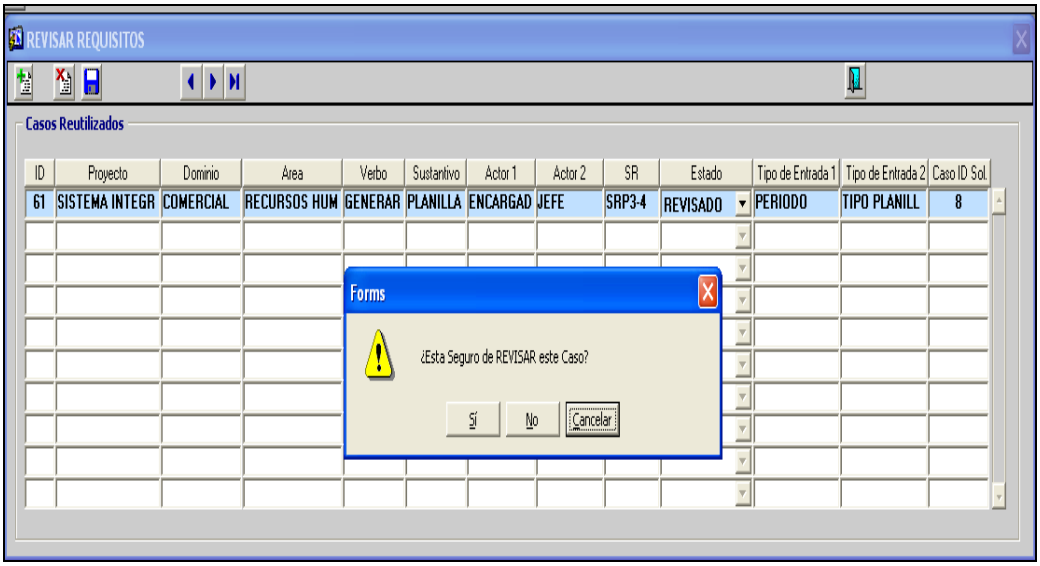


ID	Hit %	Proyecto	Dominio	Area	Verbo	Sustantivo	Actor 1	Actor 2	Tipo de Entrada 1	Tipo de Entrada 2	SR
8	89.55	SISTEMA DE PLANILLAS	COMERCIAL	RECURSOS H	GENERAR	PLANILLA	ASISTENTE	ENCARGADO	PERIODO	TIPO PLANILL	SRP3-4
9	65.25	SISTEMA DE PLANILLAS	COMERCIAL	RECURSOS HUM	GENERAR	BOLETA	ASISTENTE	ENCARGADO	PERIODO	TIPO PLANILLA	SRP3-5
23	50.25	SISTEMA DE PLANILLAS	COMERCIAL	RECURSOS HUM	GENERAR	CTS	ASISTENTE	ENCARGADO	ID EMPLEADO	FECHA	SRP3-12

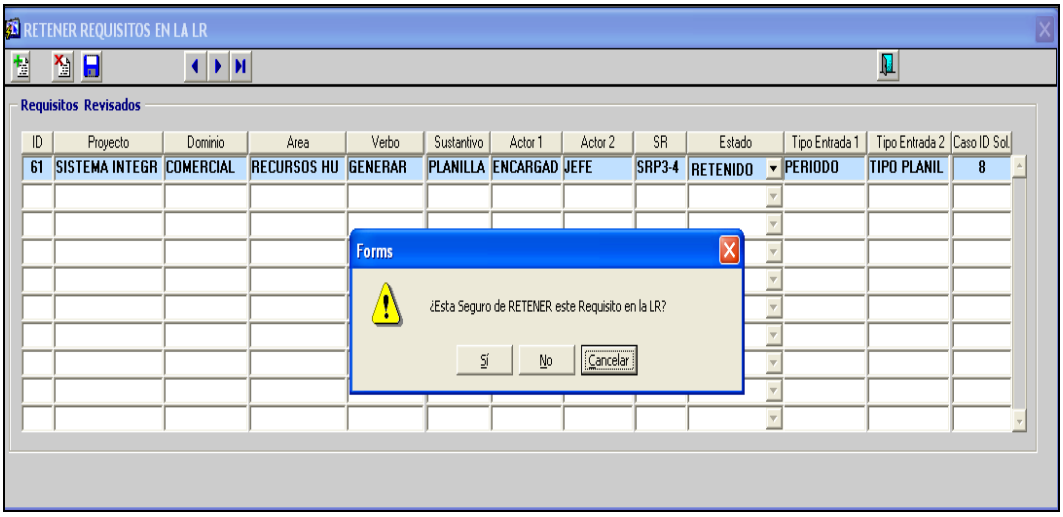
Casos similares del requisito *Generar Planilla* del caso de estudio 1, utilizando el modelo CBR propuesto.



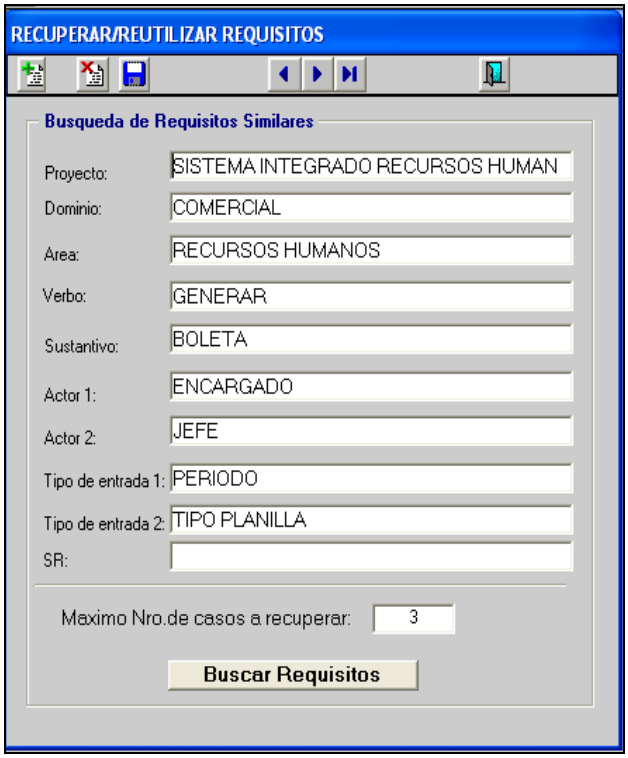
Requisito reutilizado para dar posible solución al requisito *Generar Planilla* del caso de estudio 1, utilizando el modelo CBR propuesto.



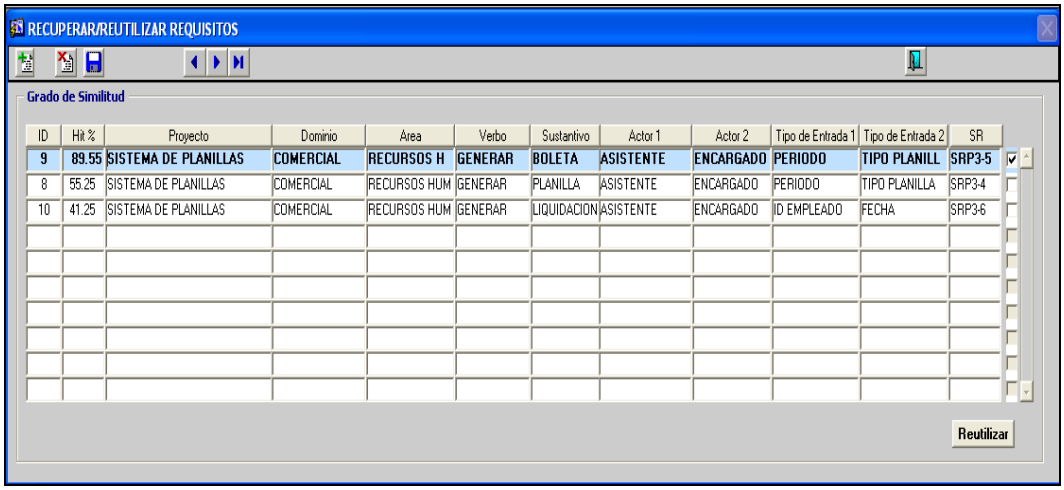
Revisión del requisito *Generar Planilla* del caso de estudio 1, utilizando el modelo CBR propuesto.



Retención del requisito G*enerar Planilla* a la librería de requisitos del caso de estudio 1, utilizando el modelo CBR propuesto.

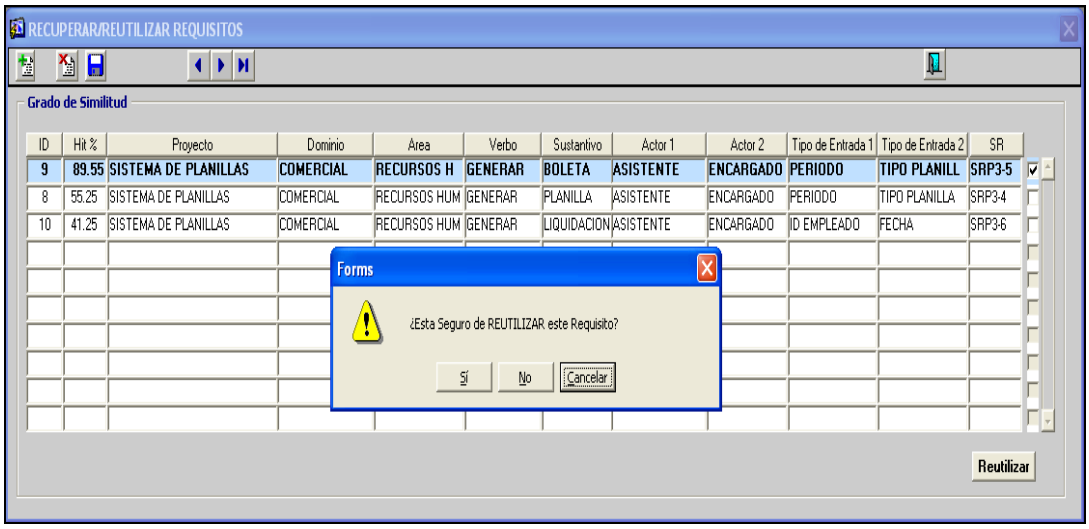


Recuperación del requisito *Generar Boleta* del caso de estudio 1, utilizando el modelo CBR propuesto.



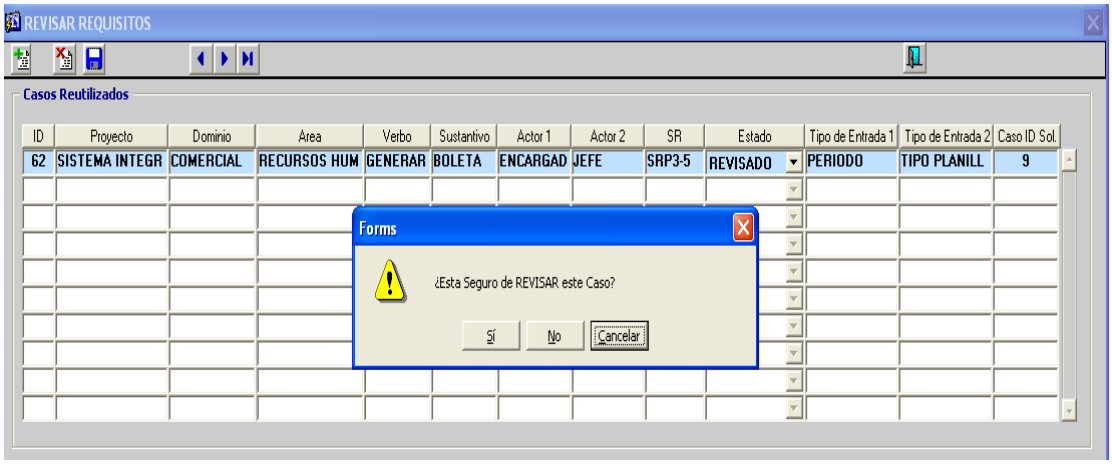
ID	Hit %	Proyecto	Dominio	Area	Verbo	Sustantivo	Actor 1	Actor 2	Tipo de Entrada 1	Tipo de Entrada 2	SR
9	89.55	SISTEMA DE PLANILLAS	COMERCIAL	RECURSOS H	GENERAR	BOLETA	ASISTENTE	ENCARGADO	PERIODO	TIPO PLANILL	SRP3-5
8	55.25	SISTEMA DE PLANILLAS	COMERCIAL	RECURSOS HUM	GENERAR	PLANILLA	ASISTENTE	ENCARGADO	PERIODO	TIPO PLANILLA	SRP3-4
10	41.25	SISTEMA DE PLANILLAS	COMERCIAL	RECURSOS HUM	GENERAR	LIQUIDACION	ASISTENTE	ENCARGADO	ID EMPLEADO	FECHA	SRP3-6

Casos similares del requisito *Generar Boleta* del caso de estudio 1, utilizando el modelo CBR propuesto.

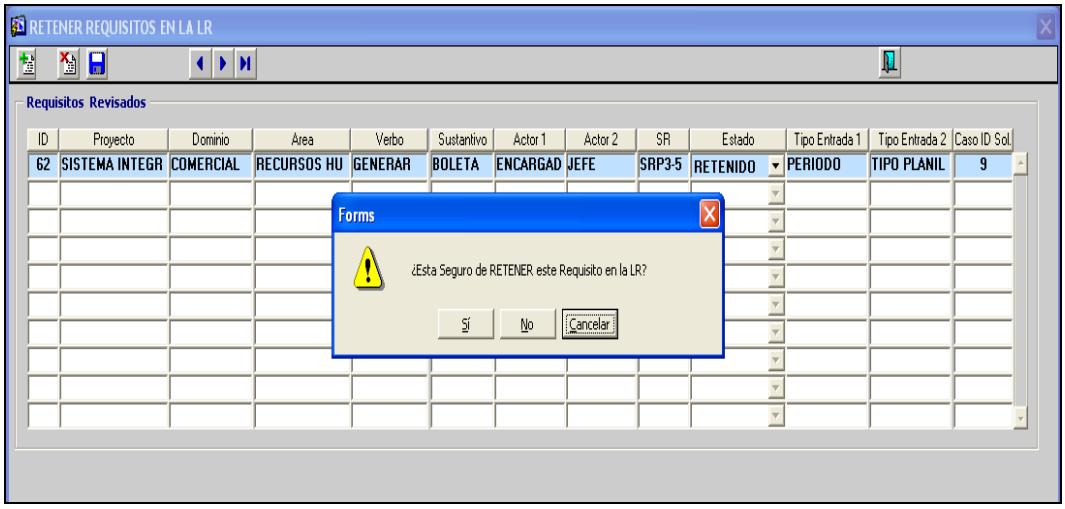


ID	Hit %	Proyecto	Dominio	Area	Verbo	Sustantivo	Actor 1	Actor 2	Tipo de Entrada 1	Tipo de Entrada 2	SR
9	89.55	SISTEMA DE PLANILLAS	COMERCIAL	RECURSOS H	GENERAR	BOLETA	ASISTENTE	ENCARGADO	PERIODO	TIPO PLANILL	SRP3-5
8	55.25	SISTEMA DE PLANILLAS	COMERCIAL	RECURSOS HUM	GENERAR	PLANILLA	ASISTENTE	ENCARGADO	PERIODO	TIPO PLANILLA	SRP3-4
10	41.25	SISTEMA DE PLANILLAS	COMERCIAL	RECURSOS HUM	GENERAR	LIQUIDACION	ASISTENTE	ENCARGADO	ID EMPLEADO	FECHA	SRP3-6

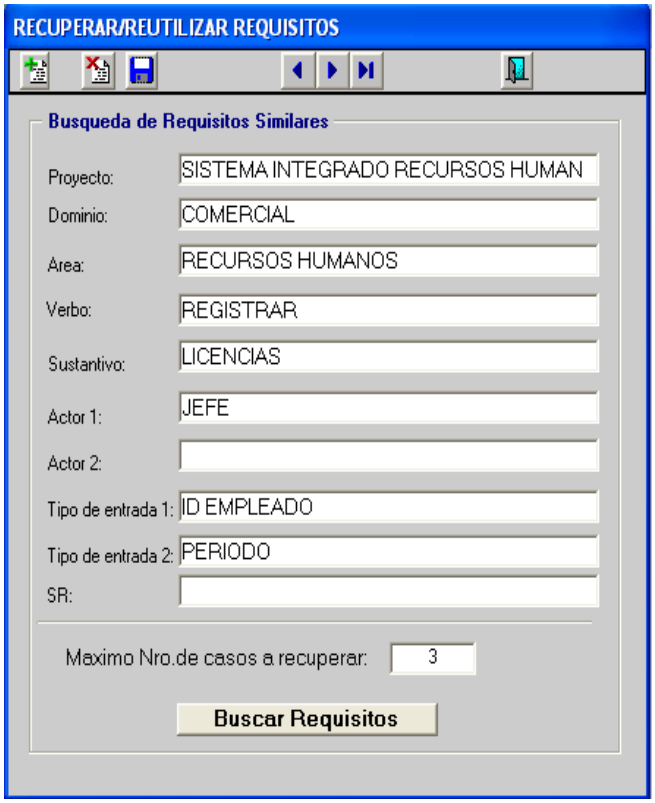
Requisito reutilizado para dar posible solución al requisito *Generar Boleta* del caso de estudio 1, utilizando el modelo CBR propuesto.



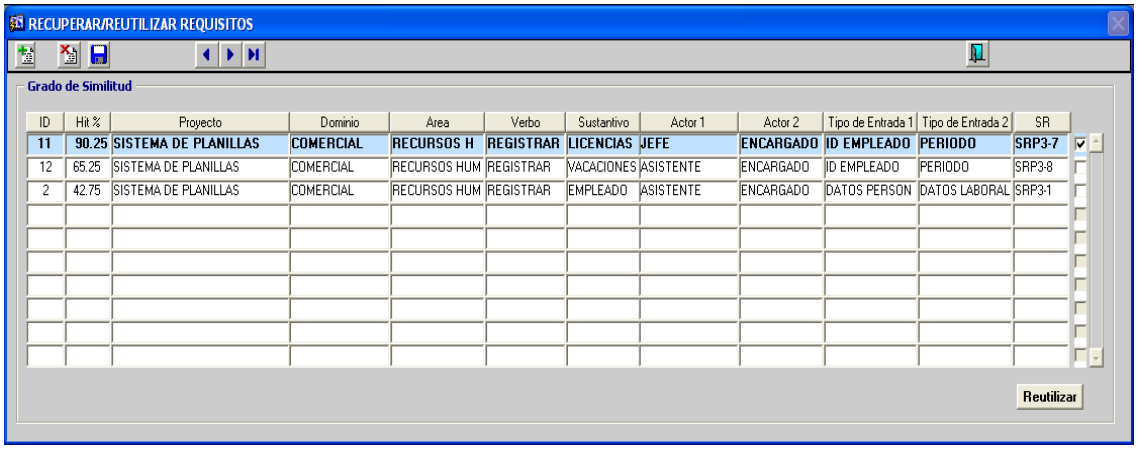
Revisión del requisito *Generar Boleta* del caso de estudio 1, utilizando el modelo CBR propuesto.



Retención del requisito G*enerar Boleta* a la librería de requisitos del caso de estudio 1, utilizando el modelo CBR propuesto.

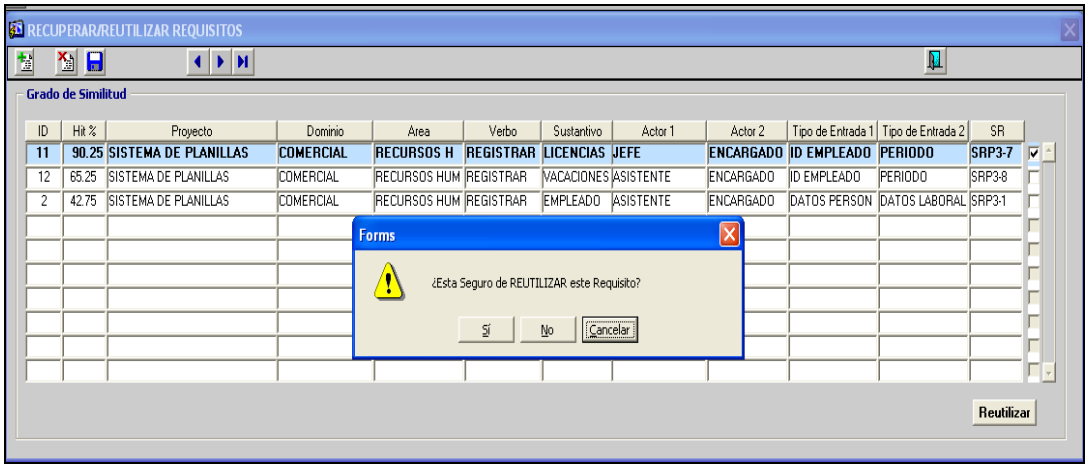


Recuperación del requisito *Registrar Licencia* del caso de estudio 1, utilizando el modelo CBR propuesto.

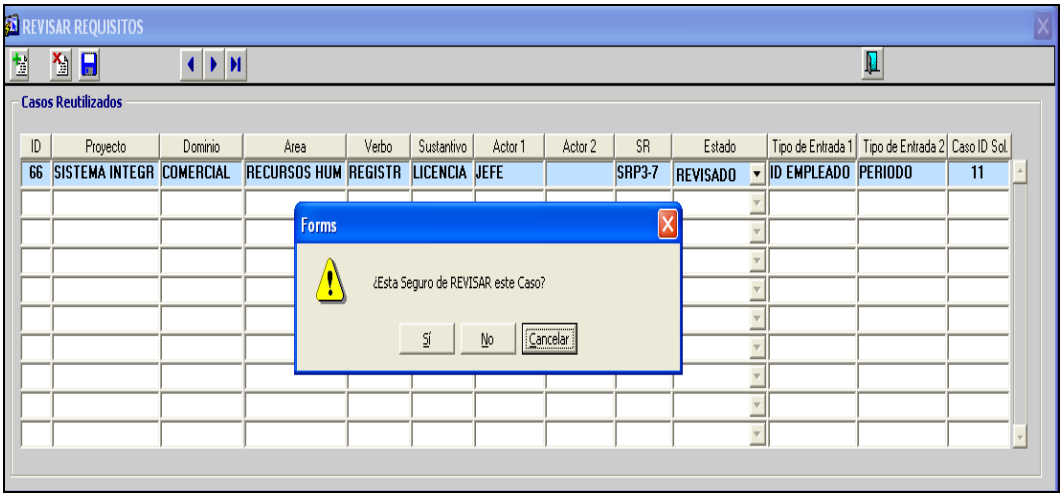


ID	Hit %	Proyecto	Dominio	Area	Verbo	Sustantivo	Actor 1	Actor 2	Tipo de Entrada 1	Tipo de Entrada 2	SR
11	90.25	SISTEMA DE PLANILLAS	COMERCIAL	RECURSOS H	REGISTRAR	LICENCIAS	JEFE	ENCARGADO	ID EMPLEADO	PERIODO	SRP3-7
12	65.25	SISTEMA DE PLANILLAS	COMERCIAL	RECURSOS HUM	REGISTRAR	VACACIONES	ASISTENTE	ENCARGADO	ID EMPLEADO	PERIODO	SRP3-8
2	42.75	SISTEMA DE PLANILLAS	COMERCIAL	RECURSOS HUM	REGISTRAR	EMPLEADO	ASISTENTE	ENCARGADO	DATOS PERSON	DATOS LABORAL	SRP3-1

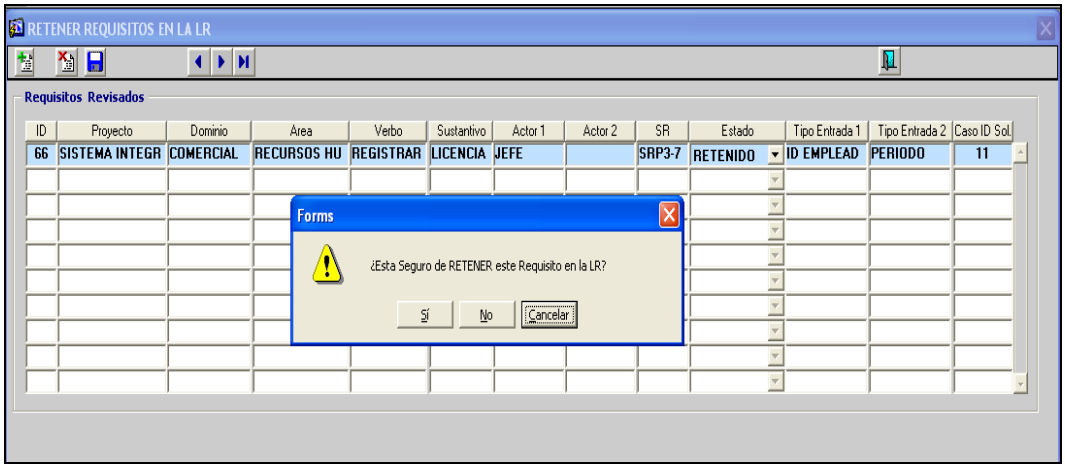
Casos similares del requisito *Registrar Licencia* del caso de estudio 1, utilizando el modelo CBR propuesto.



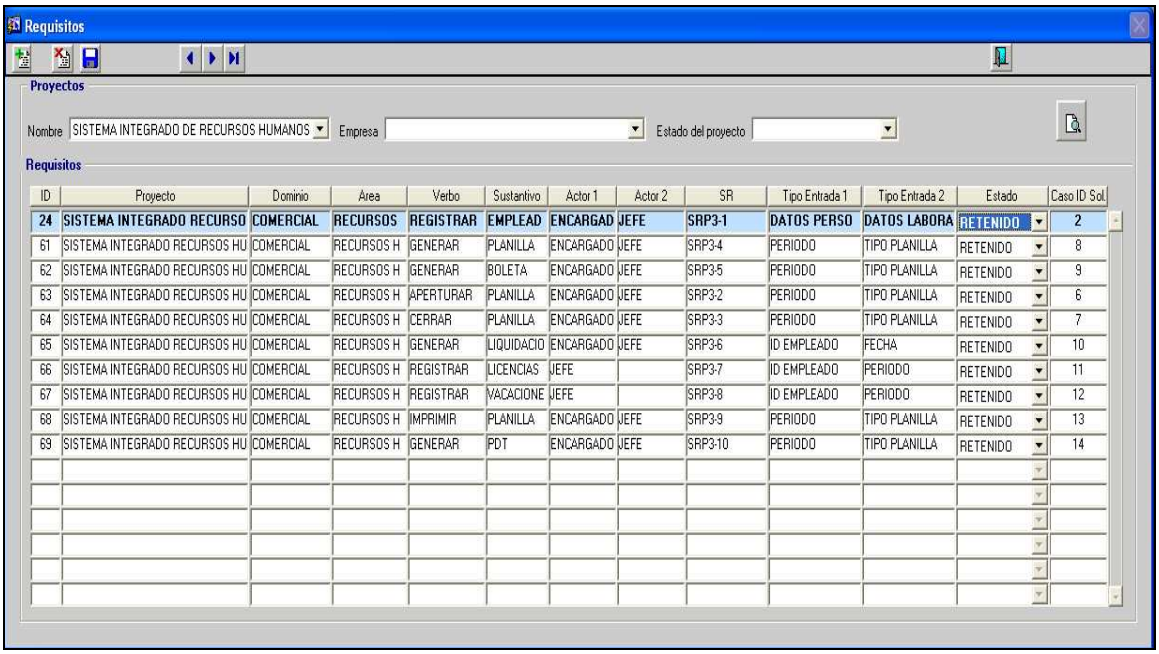
Requisito reutilizado para dar posible solución al requisito *Registrar Licencia* del caso de estudio 1, utilizando el modelo CBR propuesto.



Revisión del requisito *Registrar Licencia* del caso de estudio 1, utilizando el modelo CBR propuesto.



Retención del requisito *Registrar Licencia* a la librería de requisitos del caso de estudio 1, utilizando el modelo CBR propuesto.



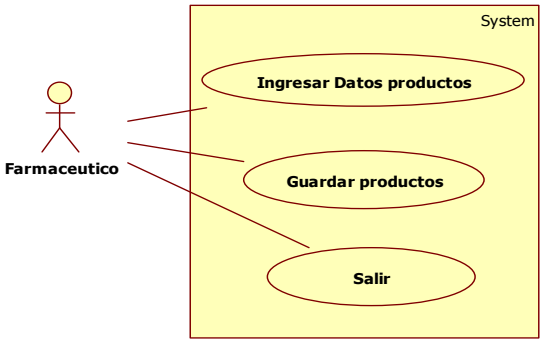
Consulta de los diez requisitos obtenidos con el modelo CBR propuesto para el caso de estudio 1: Sistema Integrado de Recursos Humanos (SIRH).

Anexo 4: Captación de requisitos usando *Casos de Uso* del caso de estudio 2: Sistema de Punto de Venta (SPTA)

Caso de uso *Registrar Productos* del caso de estudio 2: Sistema de Punto de Venta (SPTA)

Caso de Uso: R1-Registrar Productos	**Analista:** Iesus Delgado

Diagrama de Casos de Uso



Breve Descripción

El sistema deberá permitir registrar los datos de los productos farmacéuticos, tales como: código de barra, descripción resumida del producto, descripción detallada, familia, subfamilia, estado, precio de compra, precio de venta, posología, reacciones adversas, observaciones, Fecha de fabricación, Fecha de Vencimiento, fabricante, marca, distribuidor, etc.

Actores

- Farmacéutico: usuario interno a la organización que está encargado de dar asesoría relacionada a alguna posología o prescripción médica de los productos farmacéuticos a los clientes. También está destinado a realizar las ventas de los productos farmacéuticos, ya sea en el local o por Delivery. Así mismo, está destinado a dar información del producto, solicitada por los clientes.

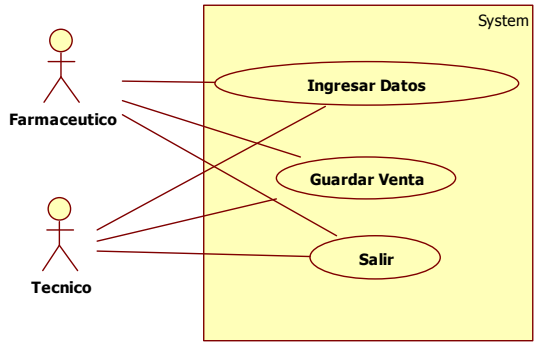
Flujo básico

- El usuario selecciona la opción "Registrar productos". El sistema muestra el formulario respectivo.
- El usuario ingresa los siguientes datos: código de barra, descripción resumida del producto, descripción detallada, familia, subfamilia, estado, precio de compra, precio de venta, posología, reacciones adversas, observaciones, Fecha de fabricación, Fecha de Vencimiento, fabricante, marca, distribuidor, etc.
- Luego de ingresar los campos obligatorios se activa el botón "Guardar".
- El sistema muestra un mensaje preguntándole si está seguro de guardar los cambios. El usuario acepta.
- Luego el sistema muestra un mensaje, especificando que los cambios se han guardado con éxito.
- El caso de uso termina.

Caso de uso *Registrar ventas* del caso de estudio 2: Sistema de Punto de Venta (SPTA)

Caso de Uso: R3-Registrar Ventas	**Analista:** Iesus Delgado

Diagrama de Casos de Uso



Breve Descripción

El sistema deberá permitir registrar las ventas realizadas por cada uno de los diferentes locales. Se debe registrar los datos generales y el detalle de las ventas, tales como: fecha, tipo de comprobante, número de comprobante, importe total, importe del IGV, cliente, productos, cantidad, precio por productos, etc. Estas ventas pueden ser hechas en el local o por Delivery.

Actores

- Farmacéutico: usuario interno a la organización que está encargado de dar asesoría relacionada a alguna posología o prescripción médica de los productos farmacéuticos a los clientes. También está destinado a realizar las ventas de los productos farmacéuticos, ya sea en el local o por Delivery. Así mismo, está destinado a dar información del producto, solicitada por los clientes.
- Técnico: usuario interno a la organización que está encargado de realizar las ventas de los productos farmacéuticos, ya sea en el local o por Delivery. También está destinado a dar información del producto, solicitada por los clientes.

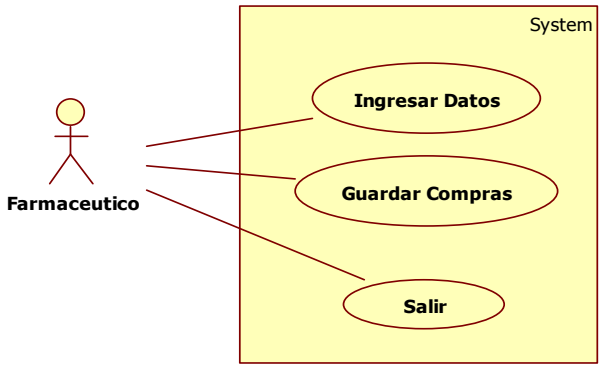
Flujo básico

- El usuario selecciona la opción “Registrar Ventas”. El sistema muestra el formulario respectivo.
- El usuario ingresa los siguientes datos: fecha, tipo de comprobante, número de comprobante, importe total, importe del IGV, cliente, productos, cantidad, precio por productos, etc.
- Luego de ingresar los campos obligatorios se activa el botón “Guardar”.
- El sistema muestra un mensaje preguntándole si está seguro de guardar los cambios. El usuario acepta.
- Luego el sistema muestra un mensaje, especificando que los cambios se han guardado con éxito.
- El caso de uso termina.

Caso de uso *Registrar Compras* del caso de estudio 2: Sistema de Punto de Venta (SPTA)

Caso de Uso: R4-Registrar Compras	**Analista:** Iesus Delgado

Diagrama de Casos de Uso



Breve Descripción

El sistema deberá permitir registrar las compras que la empresa realiza. Se debe registrar los datos generales y el detalle de las compras, tales como: fecha de compra, tipo de comprobante, número de comprobante, fecha, importe total, importe del IGV, proveedor, productos, cantidad, precio por productos, etc.

Actores

- Encargado de Compras: usuario interno a la organización que está encargado de todas las tareas relacionadas a la compras de productos, de toda la cadena de las farmacias.

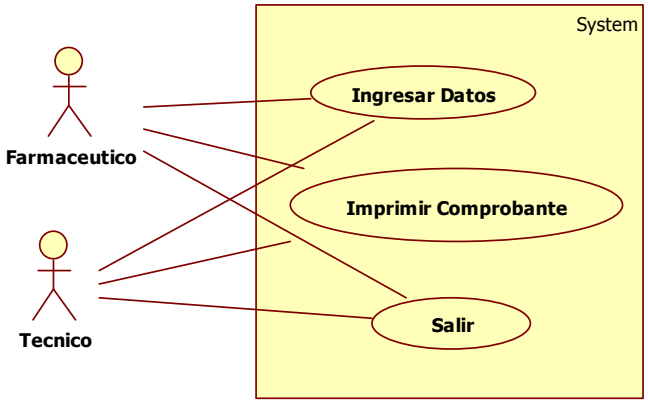
Flujo básico

- El usuario selecciona la opción “Registrar Compras”. El sistema muestra el formulario respectivo.
- El usuario ingresa los siguientes datos: fecha de compra, tipo de comprobante, número de comprobante, fecha, importe total, importe del IGV, proveedor, productos, cantidad, precio por productos, etc.
- Luego de ingresar los campos obligatorios se activa el botón “Guardar”.
- El sistema muestra un mensaje preguntándole si está seguro de guardar los cambios. El usuario acepta.
- Luego el sistema muestra un mensaje, especificando que los cambios se han guardado con éxito.
- El caso de uso termina.

Caso de uso *Generar Comprobante* del caso de estudio 2: Sistema de Punto de Venta (SPTA)

Caso de Uso: R5-Generar Comprobante	**Analista:** Iesus Delgado

Diagrama de Casos de Uso



Breve Descripción

El sistema deberá permitir generar el comprobante de pago, luego de haber realizado la venta, es decir, luego de registrar la venta de los productos farmacéuticos. Este comprobante también debe ser impreso.

Actores

- Farmacéutico: usuario interno a la organización que está encargado de dar asesoría relacionada a alguna posología o prescripción médica de los productos farmacéuticos a los clientes. También está destinado a realizar las ventas de los productos farmacéuticos, ya sea en el local o por Delivery. Así mismo, está destinado a dar información del producto, solicitada por los clientes.
- Técnico: usuario interno a la organización que está encargado de realizar las ventas de los productos farmacéuticos, ya sea en el local o por Delivery. También está destinado a dar información del producto, solicitada por los clientes.

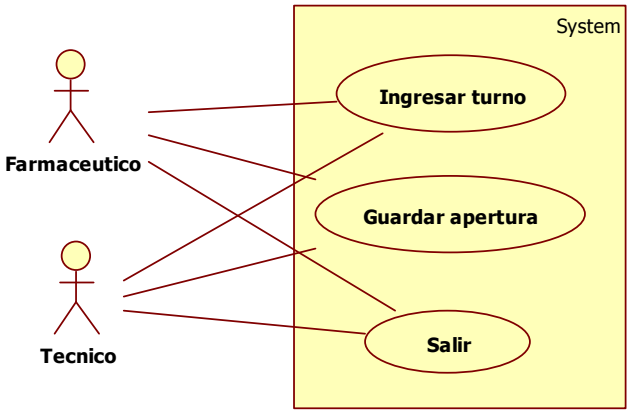
Flujo básico

- El usuario selecciona la opción “Imprimir Comprobante” que se encuentra en la pantalla “Registro de ventas”.
- El sistema muestra un mensaje preguntándole si está seguro de imprimir el comprobante.
- El usuario acepta. Luego el sistema muestra un mensaje informando que el comprobante se esta imprimiendo.
- El caso de uso termina.

Caso de uso A*perturar Caja* del caso de estudio 2: Sistema de Punto de Venta (SPTA)

Caso de Uso: R6-Aperturar Caja	**Analista:** Iesus Delgado

Diagrama de Casos de Uso



Breve Descripción

Esta funcionalidad permite abrir el día, para poder registrar todas las ventas de los productos en cada uno de los locales o farmacias. El sistema queda en el estado habilitado para el registro de las ventas. Para ello se debe ingresar el número de turno y el código del químico farmacéutico o técnico.

Actores

- Farmacéutico: usuario interno a la organización que está encargado de dar asesoría relacionada a alguna posología o prescripción médica de los productos farmacéuticos a los clientes. También está destinado a realizar las ventas de los productos farmacéuticos, ya sea en el local o por Delivery. Así mismo, está destinado a dar información del producto, solicitada por los clientes.
- Técnico: usuario interno a la organización que está encargado de realizar las ventas de los productos farmacéuticos, ya sea en el local o por Delivery. También está destinado a dar información del producto, solicitada por los clientes.

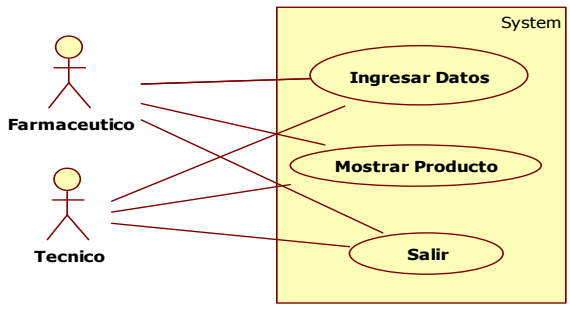
Flujo básico

- El usuario selecciona la opción “Aperturar Caja”. El sistema muestra el formulario respectivo.
- El usuario ingresa los siguientes datos: número de turno y el código del químico farmacéutico.
- Luego de ingresar los campos se activa el botón “Guardar”.
- El sistema muestra un mensaje preguntándole si está seguro de guardar los cambios. El usuario acepta.
- Luego el sistema muestra un mensaje, especificando que los cambios se han guardado con éxito.
- El caso de uso termina.

Caso de uso *Consultar Productos* del caso de estudio 2: Sistema de Punto de Venta (SPTA)

Caso de Uso: R7- Consultar Productos	**Analista:** Iesus Delgado

Diagrama de Casos de Uso



Breve Descripción

El sistema deberá permitir consultar los datos de los productos, tales como: descripción resumida del producto, descripción detallada, familia, subfamilia, estado, fecha de fabricación, fecha de vencimiento, marca, proveedor, distribuidor, precio de compra, precio de venta, posología, etc.

Actores

- Farmacéutico: usuario interno a la organización que está encargado de dar asesoría relacionada a alguna posología o prescripción médica de los productos farmacéuticos a los clientes. También está destinado a realizar las ventas de los productos farmacéuticos, ya sea en el local o por Delivery. Así mismo, está destinado a dar información del producto, solicitada por los clientes. Es el responsable del local.
- Técnico: usuario interno a la organización que está encargado de realizar las ventas de los productos farmacéuticos, ya sea en el local o por Delivery. También está destinado a dar información del producto, solicitada por los clientes.

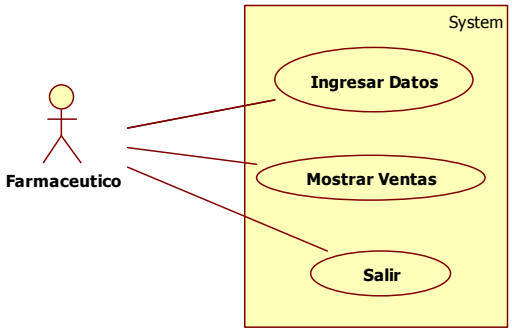
Flujo básico

- El usuario selecciona la opción “Consultar Productos”.
- El sistema muestra el formulario respectivo.
- El usuario puede consultar los productos con los siguientes filtros: Código, Familia, Subfamilia, etc.
- El sistema muestra la siguiente información de los productos: descripción resumida del producto, descripción detallada, familia, subfamilia, estado, fecha de fabricación, fecha de vencimiento, marca, proveedor, distribuidor, precio de compra, precio de venta, posología, etc.
- Luego de consultar el usuario selecciona la opción “Salir”.
- El caso de uso termina.

Caso de uso *Consultar Ventas* del caso de estudio 2: Sistema de Punto de Venta (SPTA)

Caso de Uso: R8- Consultar Ventas	**Analista:** Iesus Delgado

Diagrama de Casos de Uso



Breve Descripción

El sistema deberá permitir consultar los datos de las ventas realizadas en los locales. Los filtros de consultas se deberán ser: ventas por día, ventas por rango de fechas, ventas por tipo de comprobante, ventas por local, ventas por Delivery, productos más vendidos, etc.

Actores

- Farmacéutico: usuario interno a la organización que está encargado de dar asesoría relacionada a alguna posología o prescripción médica de los productos farmacéuticos a los clientes. También está destinado a realizar las ventas de los productos farmacéuticos, ya sea en el local o por Delivery. Así mismo, está destinado a dar información del producto, solicitada por los clientes. Es el responsable del local.

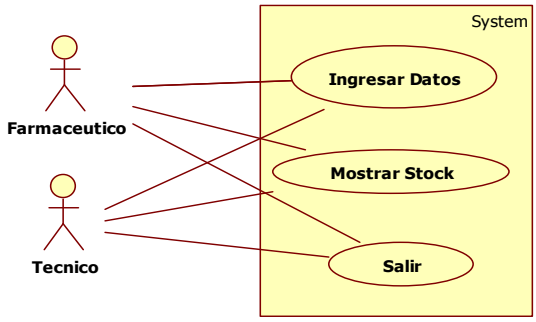
Flujo básico

- El usuario selecciona la opción “Consultar Ventas”.
- El sistema muestra el formulario respectivo.
- El usuario puede consultar los productos con los siguientes filtros: Fechas, Rango de fechas, Tipo de comprobante, por locales, por Delivery, etc.
- El sistema muestra la siguiente información de las ventas: fecha, comprobante, importe, productos, local, importe, etc.
- Luego de consultar el usuario selecciona la opción “Salir”.
- El caso de uso termina.

Caso de uso *Consultar Stock* del caso de estudio 2: Sistema de Punto de Venta (SPTA)

Caso de Uso: R9-Consultar Stock	**Analista:** Iesus Delgado

Diagrama de Casos de Uso



Breve Descripción

El sistema deberá permitir consultar el stock actual de un producto determinado, que se encuentra en una farmacia o local. Esto es necesario para saber el stock de los productos que cuenta cada local. Para ello bastará con ingresar el código del producto.

Actores

- Farmacéutico: usuario interno a la organización que está encargado de dar asesoría relacionada a alguna posología o prescripción médica de los productos farmacéuticos a los clientes. También está destinado a realizar las ventas de los productos farmacéuticos, ya sea en el local o por Delivery. Así mismo, está destinado a dar información del producto, solicitada por los clientes. Es el responsable del local.
- Técnico: usuario interno a la organización que está encargado de realizar las ventas de los productos farmacéuticos, ya sea en el local o por Delivery. También está destinado a dar información del producto, solicitada por los clientes.

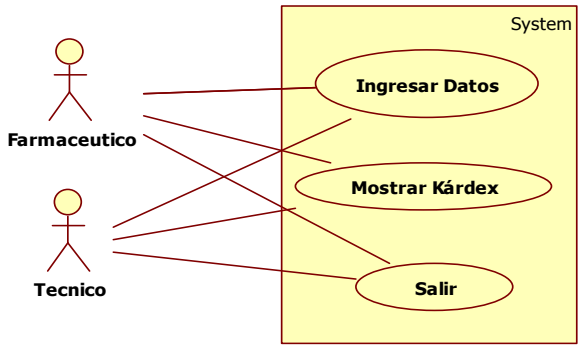
Flujo básico

- El usuario selecciona la opción “Consultar Stock”.
- El sistema muestra el formulario respectivo.
- El usuario ingresa el Código del producto.
- El sistema muestra el stock actual del producto.
- Luego de consultar el usuario selecciona la opción “Salir”.
- El caso de uso termina.

Caso de uso *Consultar Kárdex* del caso de estudio 2: Sistema de Punto de Venta (SPTA)

Caso de Uso: R10-Consultar Kárdex	**Analista:** Iesus Delgado

Diagrama de Casos de Uso



Breve Descripción

El sistema deberá permitir consultar el kárdex por producto, esto permitirá saber cuál ha sido el movimiento del producto; el ingreso y salida del producto, cantidad, fecha, origen, etc. Para ello bastará con ingresar el código del producto.

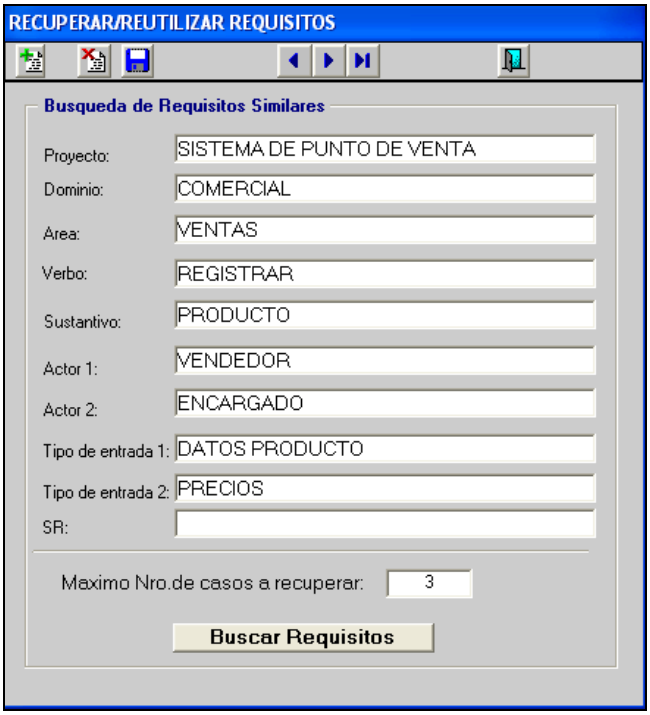
Actores

- Farmacéutico: usuario interno a la organización que está encargado de dar asesoría relacionada a alguna posología o prescripción médica de los productos farmacéuticos a los clientes. También está destinado a realizar las ventas de los productos farmacéuticos, ya sea en el local o por Delivery. Así mismo, está destinado a dar información del producto, solicitada por los clientes. Es el responsable del local.
- Técnico: usuario interno a la organización que está encargado de realizar las ventas de los productos farmacéuticos, ya sea en el local o por Delivery. También está destinado a dar información del producto, solicitada por los clientes.

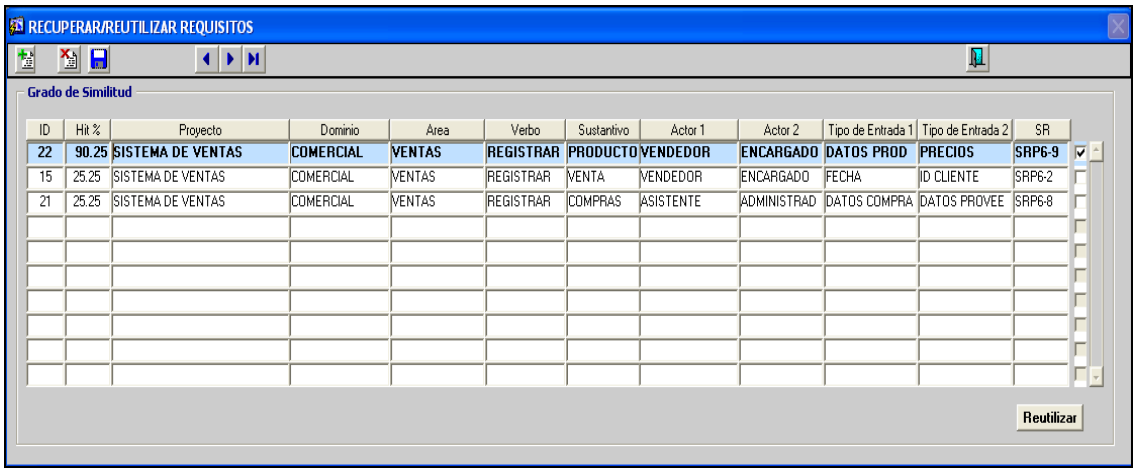
Flujo básico

- El usuario selecciona la opción “Consultar Kárdex”.
- El sistema muestra el formulario respectivo.
- El usuario ingresa el Código del producto.
- El sistema muestra el moviendo del producto: fecha de ingreso de mercadería, cantidad, origen, fecha de salidas de mercadería, cantidad, origen, etc.
- Luego de consultar el usuario selecciona la opción “Salir”.
- El caso de uso termina.

Anexo 5: Captación de requisitos usando *el modelo CBR propuesto* del caso de estudio 2: Sistema de Punto de Venta (SPTA)

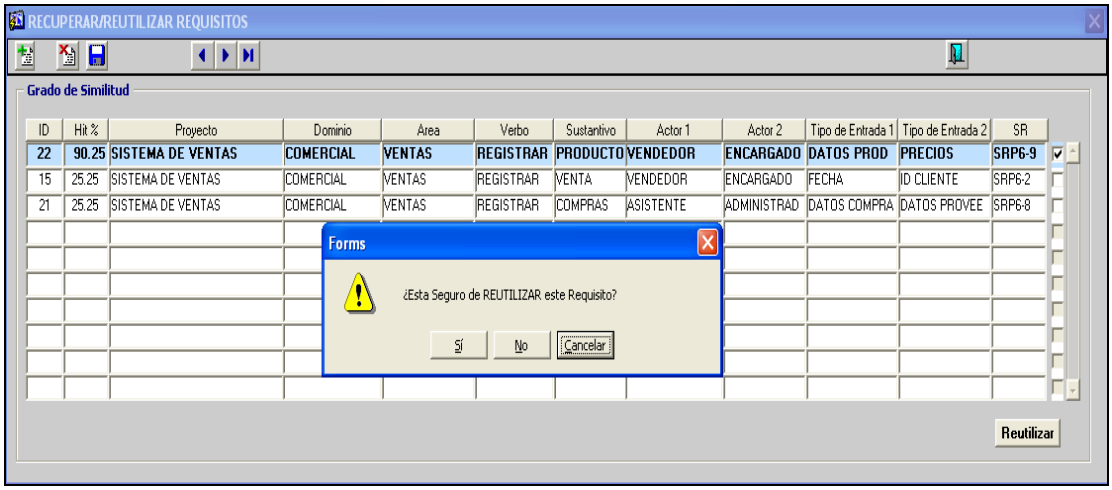


Recuperación del requisito *Registrar Producto* del caso de estudio 2, utilizando el modelo CBR propuesto.

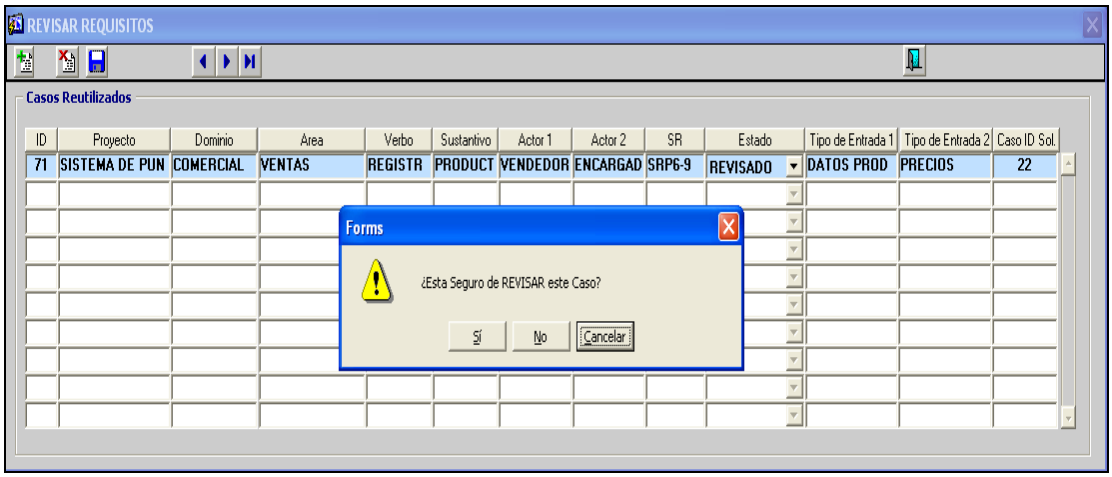


ID	Hit %	Proyecto	Dominio	Area	Verbo	Sustantivo	Actor 1	Actor 2	Tipo de Entrada 1	Tipo de Entrada 2	SR
22	90.25	SISTEMA DE VENTAS	COMERCIAL	VENTAS	REGISTRAR	PRODUCTO	VENDEDOR	ENCARGADO	DATOS PROD	PRECIOS	SRP6-9
15	25.25	SISTEMA DE VENTAS	COMERCIAL	VENTAS	REGISTRAR	VENTA	VENDEDOR	ENCARGADO	FECHA	ID CLIENTE	SRP6-2
21	25.25	SISTEMA DE VENTAS	COMERCIAL	VENTAS	REGISTRAR	COMPRAS	ASISTENTE	ADMINISTRAD	DATOS COMPRA	DATOS PROVEE	SRP6-8

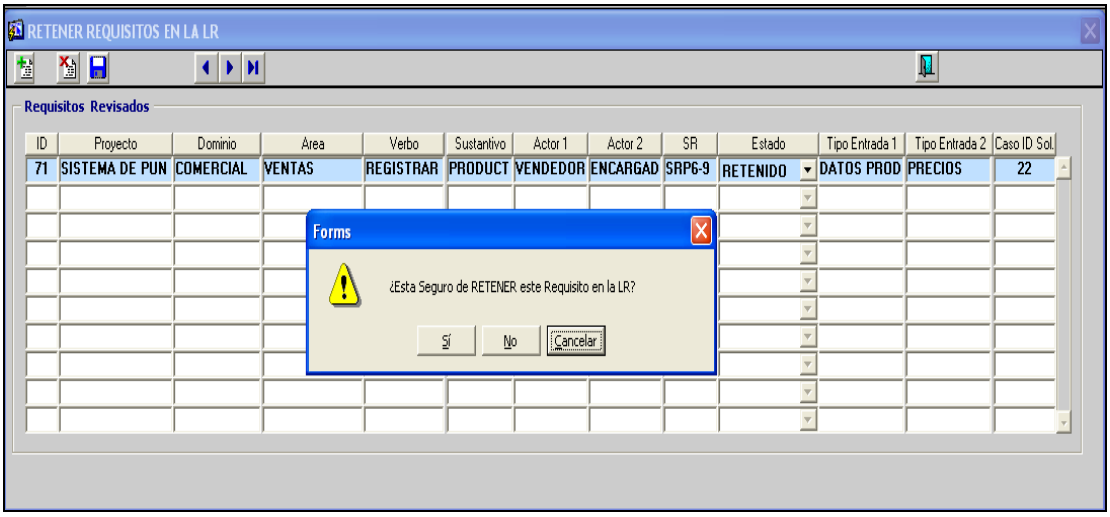
Casos similares del requisito *Registrar Productos* del caso de estudio 2, utilizando el modelo CBR propuesto.



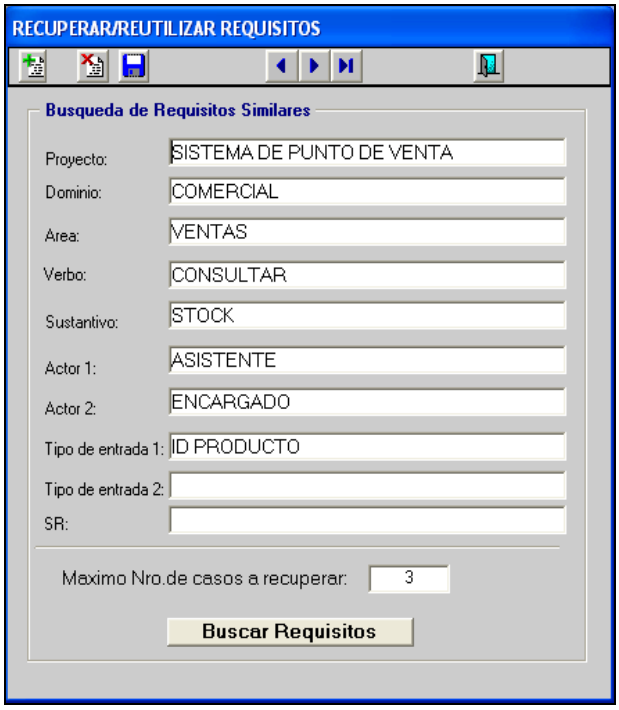
Requisito reutilizado para dar posible solución al requisito *Registrar Producto* del caso de estudio 2, utilizando el modelo CBR propuesto.



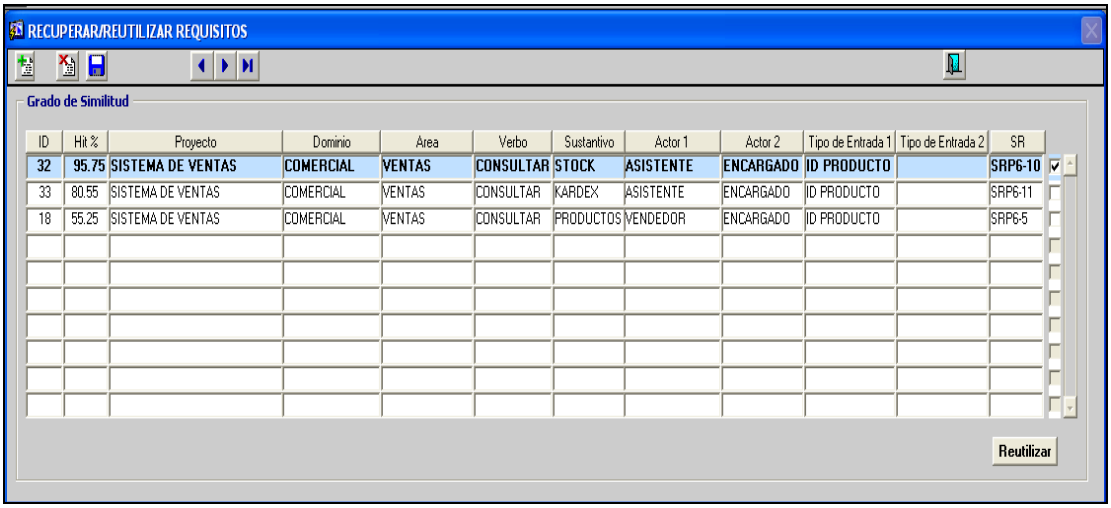
Revisión del requisito *Registrar Producto* del caso de estudio 2, utilizando el modelo CBR propuesto.



Retención del requisito *Registrar Producto* a la librería de requisitos del caso de estudio 2, utilizando el modelo CBR propuesto.

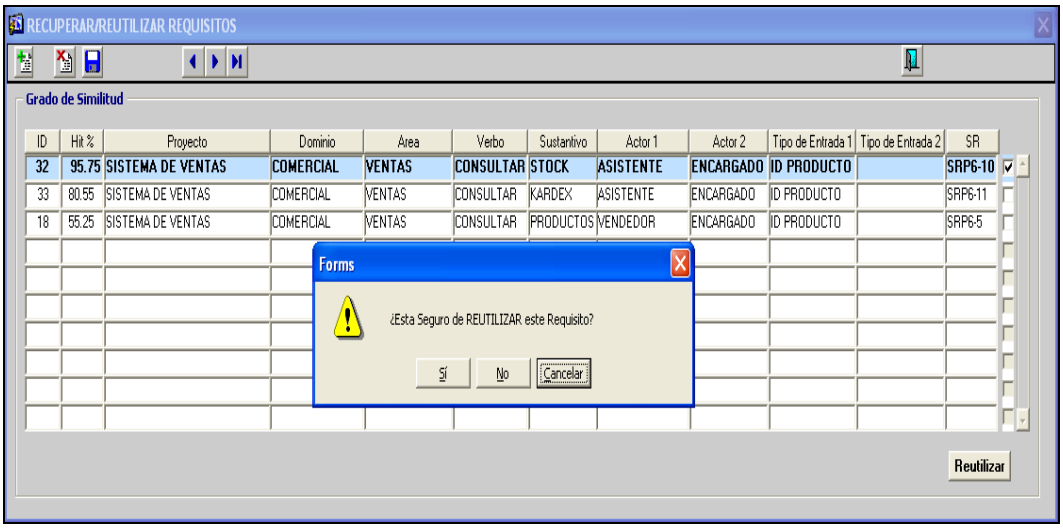


Recuperación del requisito *Consultar Stock* del caso de estudio 2, utilizando el modelo CBR propuesto.



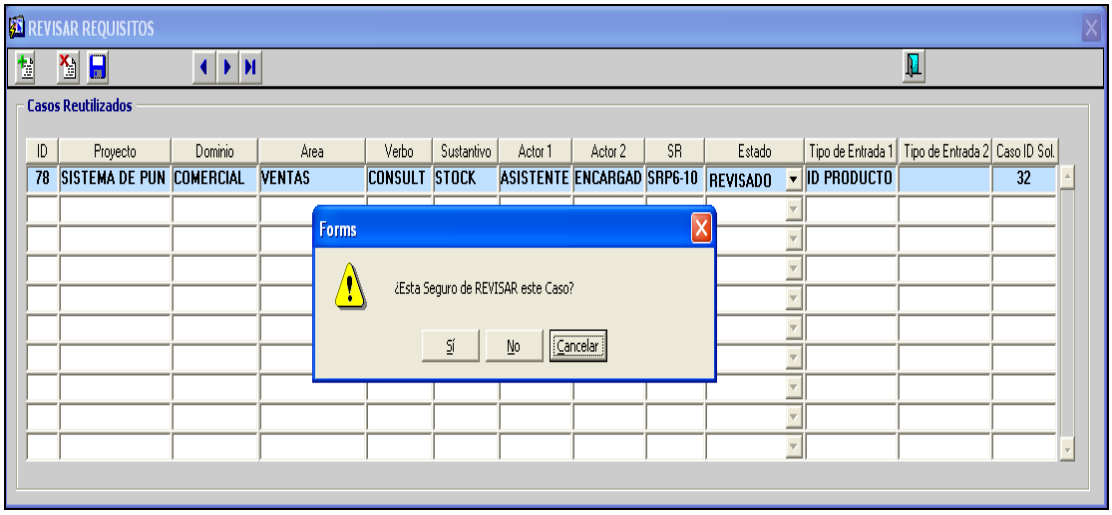
ID	Hit %	Proyecto	Dominio	Area	Verbo	Sustantivo	Actor 1	Actor 2	Tipo de Entrada 1	Tipo de Entrada 2	SR
32	95.75	SISTEMA DE VENTAS	COMERCIAL	VENTAS	CONSULTAR	STOCK	ASISTENTE	ENCARGADO	ID PRODUCTO		SRP6-10
33	80.55	SISTEMA DE VENTAS	COMERCIAL	VENTAS	CONSULTAR	KARDEX	ASISTENTE	ENCARGADO	ID PRODUCTO		SRP6-11
18	55.25	SISTEMA DE VENTAS	COMERCIAL	VENTAS	CONSULTAR	PRODUCTOS	VENDEDOR	ENCARGADO	ID PRODUCTO		SRP6-5

Casos similares del requisito *Consultar Stock* del caso de estudio 2, utilizando el modelo CBR propuesto.

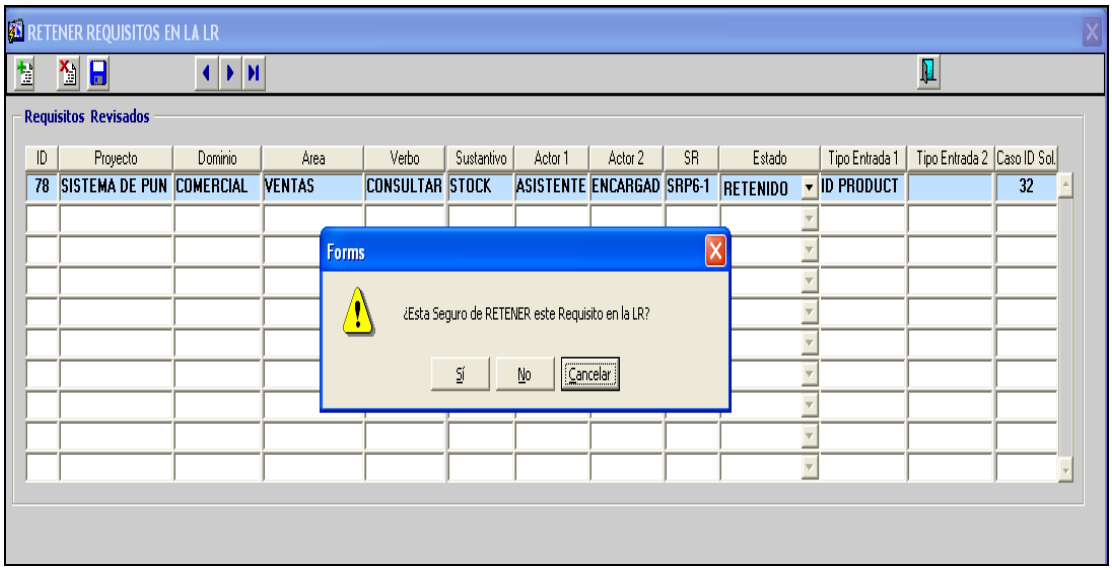


ID	Hit %	Proyecto	Dominio	Area	Verbo	Sustantivo	Actor 1	Actor 2	Tipo de Entrada 1	Tipo de Entrada 2	SR
32	95.75	SISTEMA DE VENTAS	COMERCIAL	VENTAS	CONSULTAR	STOCK	ASISTENTE	ENCARGADO	ID PRODUCTO		SRP6-10
33	80.55	SISTEMA DE VENTAS	COMERCIAL	VENTAS	CONSULTAR	KARDEX	ASISTENTE	ENCARGADO	ID PRODUCTO		SRP6-11
18	55.25	SISTEMA DE VENTAS	COMERCIAL	VENTAS	CONSULTAR	PRODUCTOS	VENDEDOR	ENCARGADO	ID PRODUCTO		SRP6-5

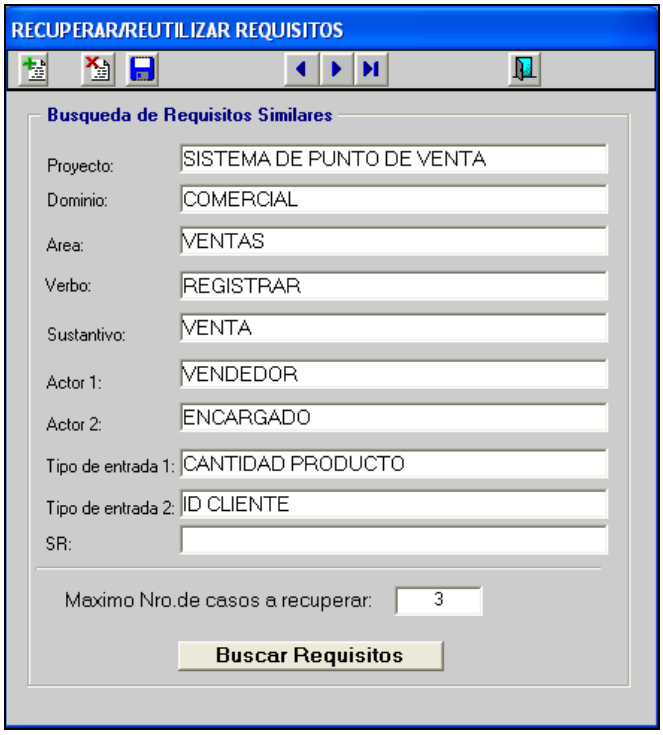
Requisito reutilizado para dar posible solución al requisito *Consultar Stock* del caso de estudio 2, utilizando el modelo CBR propuesto.



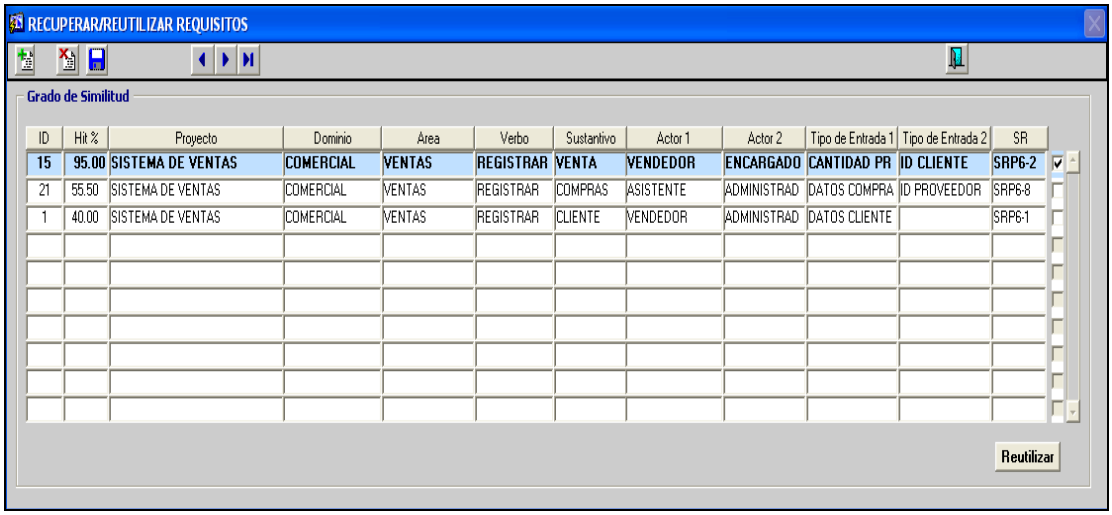
Revisión del requisito *Consultar Stock* del caso de estudio 2, utilizando el modelo CBR propuesto.



Retención del requisito *Consultar Stock* a la librería de requisitos del caso de estudio 2, utilizando el modelo CBR propuesto.

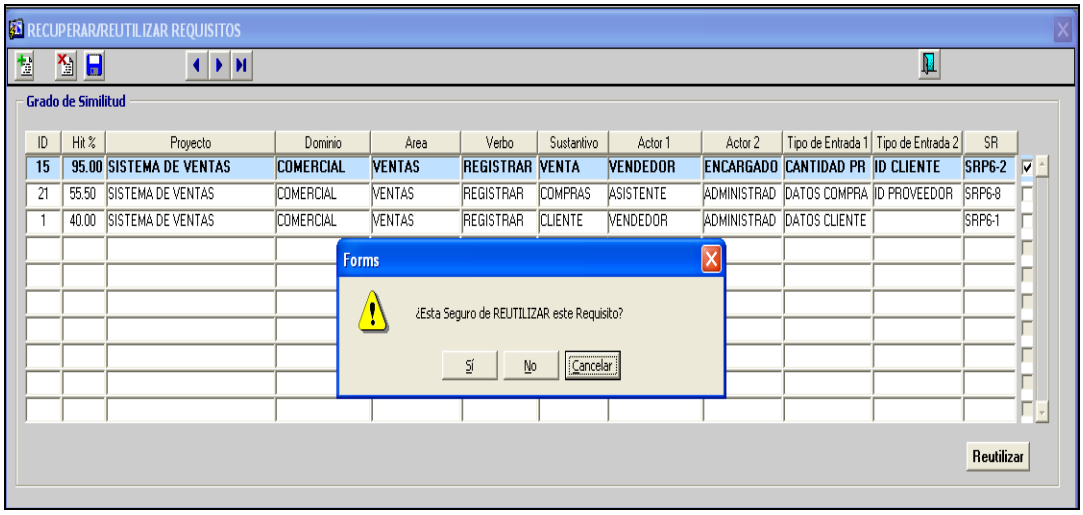


Recuperación del requisito *Registrar Venta* del caso de estudio 2, utilizando el modelo CBR propuesto.

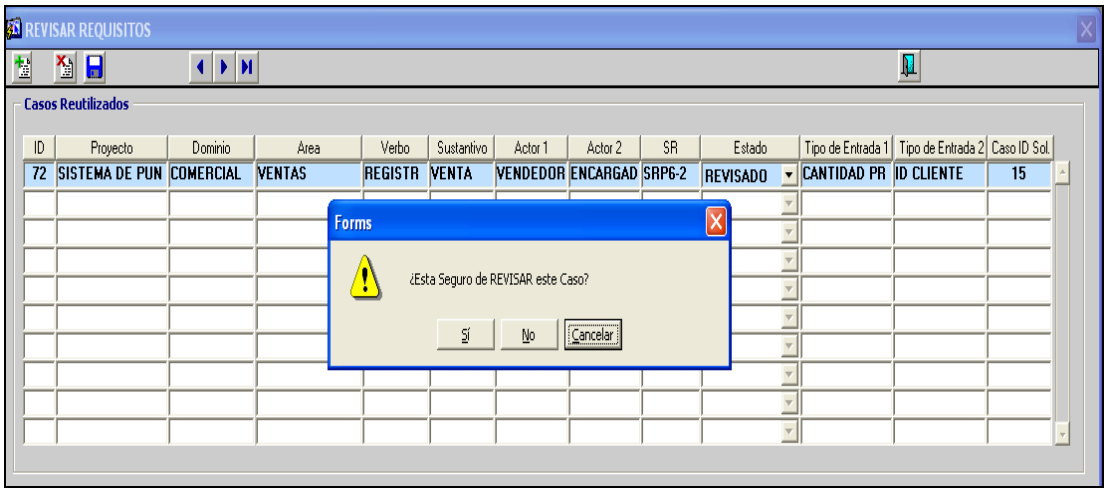


ID	Hit %	Proyecto	Dominio	Area	Verbo	Sustantivo	Actor 1	Actor 2	Tipo de Entrada 1	Tipo de Entrada 2	SR
15	95.00	SISTEMA DE VENTAS	COMERCIAL	VENTAS	REGISTRAR	VENTA	VENDEDOR	ENCARGADO	CANTIDAD PR	ID CLIENTE	SRP6-2
21	55.50	SISTEMA DE VENTAS	COMERCIAL	VENTAS	REGISTRAR	COMPRAS	ASISTENTE	ADMINISTRAD	DATOS COMPRA	ID PROVEEDOR	SRP6-8
1	40.00	SISTEMA DE VENTAS	COMERCIAL	VENTAS	REGISTRAR	CLIENTE	VENDEDOR	ADMINISTRAD	DATOS CLIENTE		SRP6-1

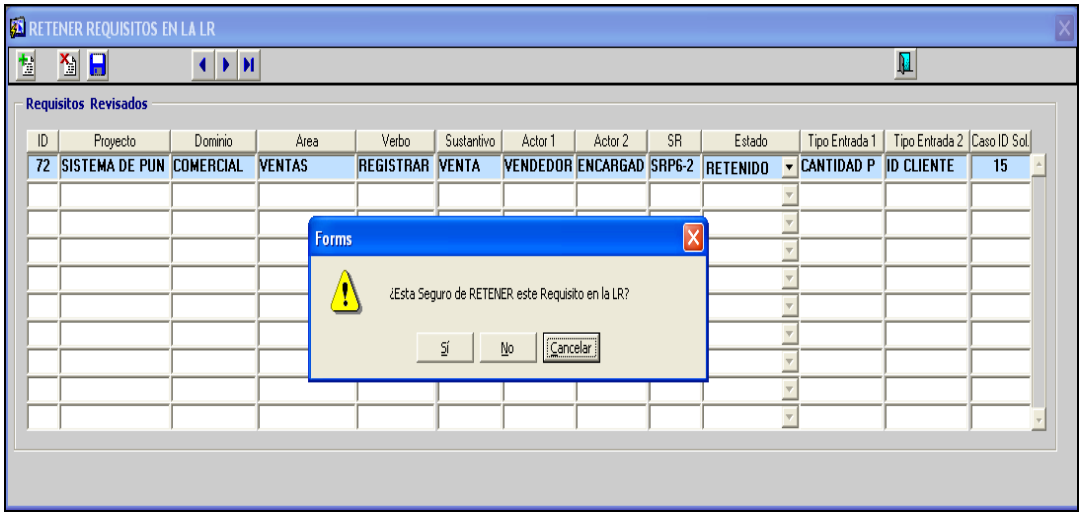
Casos similares del requisito *Registrar Venta* del caso de estudio 2, utilizando el modelo CBR propuesto.



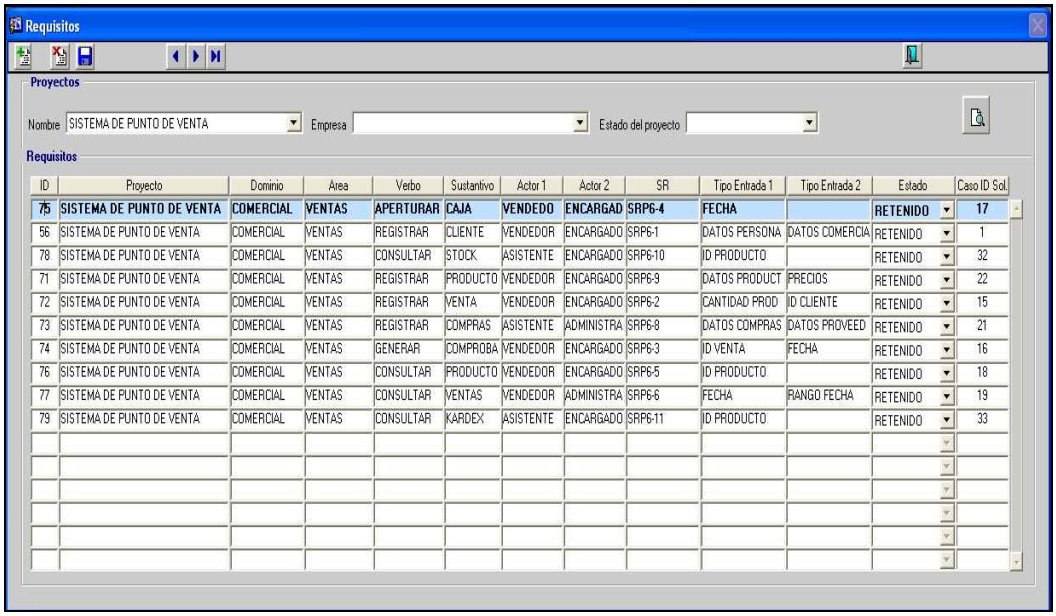
Requisito reutilizado para dar posible solución al requisito *Registrar Venta* del caso de estudio 2, utilizando el modelo CBR propuesto.



Revisión del requisito *Registrar Venta* del caso de estudio 2, utilizando el modelo CBR propuesto.



Retención del requisito *Registrar Venta* a la librería de requisitos del caso de estudio 2, utilizando el modelo CBR propuesto.



Consulta de los diez requisitos obtenidos con el modelo CBR propuesto para el caso de estudio 2: Sistema de Punto de Venta (SPTA).

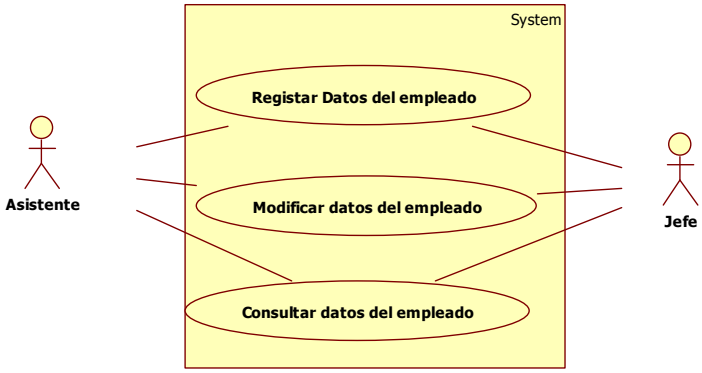
Anexo 6: Resumen del documento *Especificación de Requisito SRP3-1*

Documento de Especificación de Requisitos

Requisito: Registrar Empleado

Nombre del documento: SR3-1.doc

I. Diagrama de Casos de Uso



II. Especificaciones Funcionales

(a) Breve Descripción

Este Caso de Uso se refiere a las actividades llevadas a cabo para completar satisfactoriamente el registro de los datos de los empleados, el cual puede ser realizado por el Jefe del Recursos Humanos o por el Asistente de Recursos Humanos. El actor podrá registrar los datos del empleado, modificar y consultar su información.

(b) Actores

Jefe de Recursos Humanos: usuario interno a la organización que está encargado de velar por el cumplimiento de las actividades realizadas con el personal que trabaja en la empresa.

Asistente de Recursos Humanos: usuario interno a la organización que está encargado de ayudar en las labores y actividades relacionadas con el personal que trabaja en la empresa.

(c) Precondiciones

El actor cuenta con una cuenta de usuario registrada en el sistema para el acceso al mismo, el cual le permitirá realizar todas las operaciones relacionadas con los empleados de la empresa de acuerdo a sus roles.

(d) Post-Condiciones

Se tienen registrados los datos personales del empleado.

(e) Flujo Básico

1. El sistema muestra la pantalla de Login para acceder al Sistema.
2. El Encargado de Personal ingresa los datos de autenticación para ingresar al sistema.
3. El sistema inicia la sesión del usuario o actor y le presenta la pantalla de Registro de datos con la pestaña “Datos Personales” habilitado.

4. El actor procede con el llenado de los campos solicitados por el Sistema, los mismos que deberá llenar antes de proceder con los demás pasos. La información que deberá incluir en este rubro consiste en:
   - Estado
   - Apellido Paterno
   - Apellido Materno
   - Nombres
   - Sexo (Género)
   - Tipo Documento
   - Número de Documento de Identidad
   - Número de Documento
   - Estado Civil
   - Número de teléfono
   - Número de teléfono móvil
   - Dirección
   - Departamento
   - Provincia
   - Distrito
5. El actor presiona el botón “Guardar” para guardar los datos ingresados.

(f) Flujos Alternativos

[FA1] EL actor escoge la pestaña “Datos Laborales”

1. El Encargado de Personal elige la pestaña “Laboral” para proceder con el registro de los datos laborales.
2. El actor procede con el llenado de la información referida a sus a los datos laborales, entre los cuales se puede mencionar:

   - Tipo de Trabajador (empleado/obrero)
   - Fecha de Ingreso
   - Cargo
   - Área de la empresa a la que pertenece

- Fecha de Condición
- Tipo de AFP
- Código de AFP
- Fecha de ingreso a la AFP
- Motivo de Cese
- Fecha de cese
- Tipo de Carnet de EsSalud
- Código de carnet de EsSalud
- Pertenece a EsSalud Vida

3. El actor presiona el botón “Guardar” para guardar los datos ingresados.

[FA2] EL Encargado de Personal escoge la pestaña “Remuneraciones”

1. El actor elige la pestaña “Remuneraciones” para proceder con el registro de los datos correspondientes a los datos del cálculo de su planilla.
2. El sistema muestra la pantalla para el registro de la Remuneraciones del empleado.
3. El actor procede con el llenado de la información referida a Remuneraciones, entre los cuales se podemos mencionar:
   - Concepto de ingresos (Básico, bonificaciones, movilidad, etc.)
   - Importe
   - Tipo de moneda
   - Estado
   - Fecha de vigencia
4. Encargado de Personal presiona el botón “Guardar” para guardar los datos ingresados.

[FA3] El actor decide salir del sistema sin registrar sus datos

1. El actor no ingresa ningún dato y sale de la pantalla donde se encuentra.

2. El caso de uso termina.

(g) Requerimientos Especiales

No aplica.

(h) Puntos de Inclusión

No aplica.

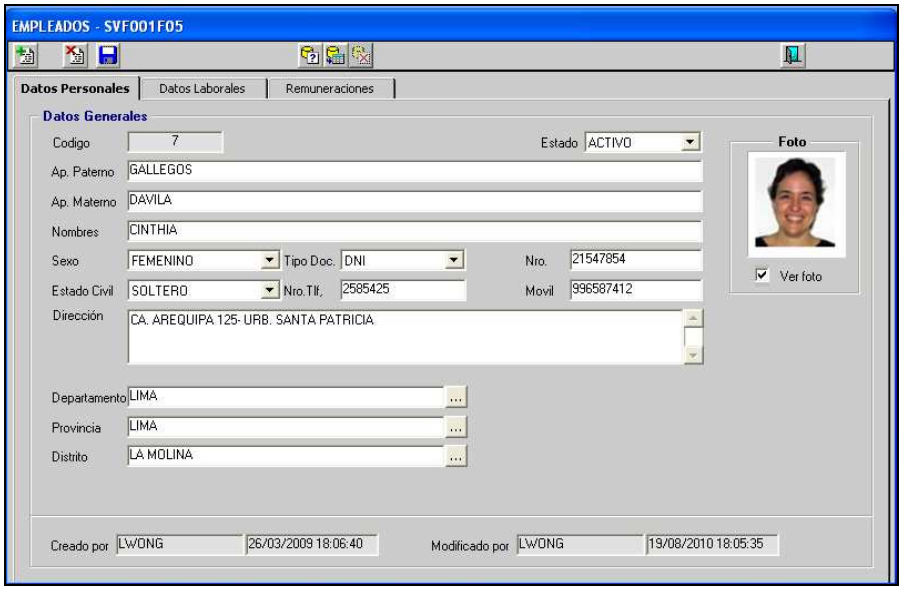
(i) Puntos de Extensión/Herencia

No aplica.

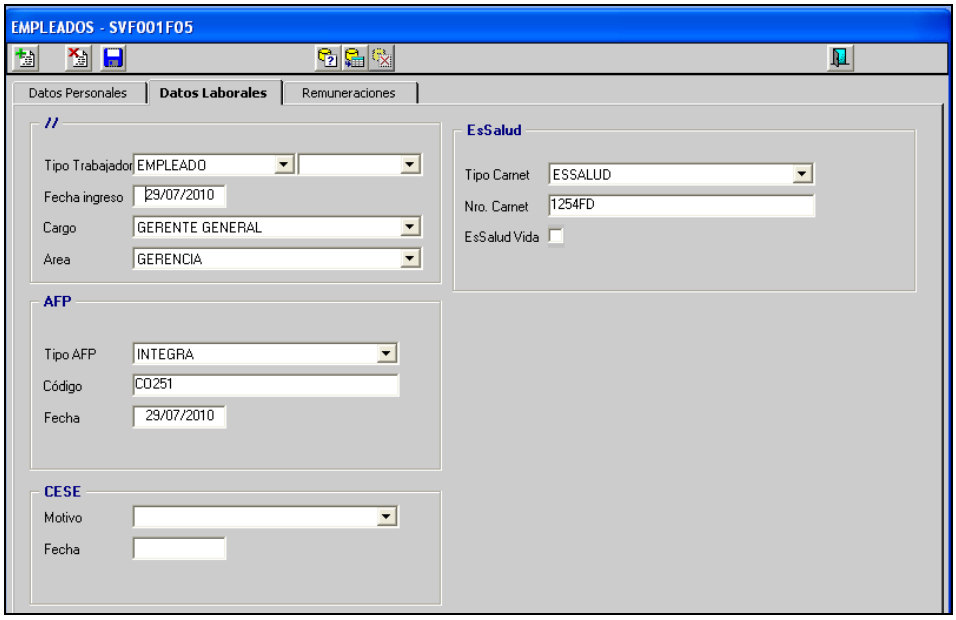
(j) Excepciones

No aplica.

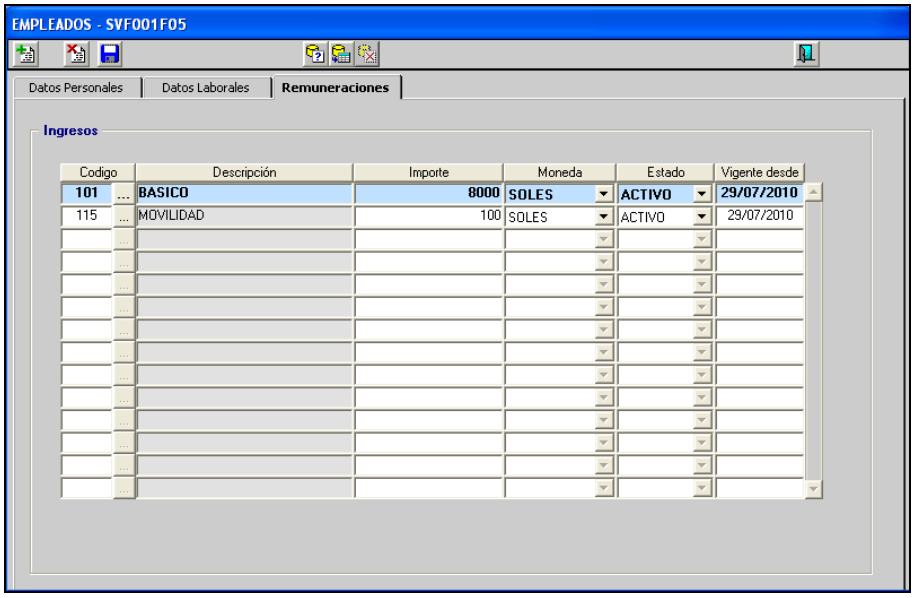
(k) Prototipos del Sistema



Pestaña de registro de Datos Personales.



Pestaña de registro de Datos Laborales.



Pestaña de registro de Datos sobre remuneraciones.

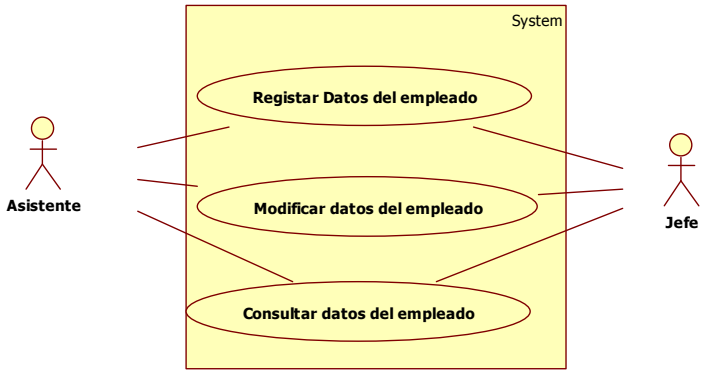
Anexo 7: Resumen del documento *Especificación de Requisito SRP3-1C*

Documento de Especificación de Requisitos

Requisito: Registrar Empleado

Nombre del documento: SR3-1C.doc

I. Diagrama de Casos de Uso



II. Especificaciones Funcionales

(a) Breve Descripción

Este Caso de Uso se refiere a las actividades llevadas a cabo para completar satisfactoriamente el registro de los datos de los empleados, el cual puede ser realizado por el Jefe del Recursos Humanos o por el Asistente de Recursos

Humanos. El actor podrá registrar los datos del empleado, modificar y consultar su información.

(b) ***Actores***

Jefe de Recursos Humanos: usuario interno a la organización que está encargado de velar por el cumplimiento de las actividades realizadas con el personal que trabaja en la empresa.

Asistente de Recursos Humanos: usuario interno a la organización que está encargado de ayudar en las labores y actividades relacionadas con el personal que trabaja en la empresa.

(c) Precondiciones

El actor cuenta con una cuenta de usuario registrada en el sistema para el acceso al mismo, el cual le permitirá realizar todas las operaciones relacionadas con los empleados de la empresa de acuerdo a sus roles.

(d) Post-Condiciones

Se tienen registrados los datos personales del empleado.

(e) Flujo Básico

1. El sistema muestra la pantalla de Login para acceder al Sistema.
2. Encargado de Personal ingresa los datos de autenticación para ingresar al sistema.
3. El sistema inicia la sesión del usuario o actor y le presenta la pantalla de Registro de datos con la pestaña “Datos Personales” habilitado.
4. El actor procede con el llenado de los campos solicitados por el Sistema, los mismos que deberá llenar antes de proceder con los demás pasos. La información que deberá incluir en este rubro consiste en:

- Estado
- Apellido Paterno
- Apellido Materno
- Nombres
- Sexo (Género)
- Tipo Documento
- Número de Documento de Identidad
- Número de Documento
- Estado Civil
- Número de teléfono
- Número de teléfono móvil
- Dirección
- Departamento
- Provincia
- Distrito

5. El actor presiona el botón “Guardar” para guardar los datos ingresados.

(f) Flujos Alternativos

[FA1] EL actor escoge la pestaña “Datos Laborales”

1. El Encargado de Personal elige la pestaña “Laboral” para proceder con el registro de los datos laborales.
2. El actor procede con el llenado de la información referida a sus a los datos laborales, entre los cuales se puede mencionar:

- Tipo de Trabajador (empleado/obrero)
- Fecha de Ingreso
- Cargo
- Área de la empresa a la que pertenece
- Fecha de Condición

- Tipo de AFP
- Código de AFP
- Fecha de ingreso a la AFP
- Motivo de Cese
- Fecha de cese
- Tipo de Carnet de EsSalud
- Código de carnet de EsSalud
- Indicador (Pertenece a EsSalud Vida)
- Forma de Pago:
    - Tipo (efectivo, cheque, depósito, etc.)
    - Banco
    - Tipo de Cuenta
    - Número de Cuenta

3. El actor presiona el botón “Guardar” para guardar los datos ingresados.

[FA2] EL Encargado de Personal escoge la pestaña “Remuneraciones”

1. El actor elige la pestaña “Remuneraciones” para proceder con el registro de los datos correspondientes a los datos del cálculo de su planilla.
2. El sistema muestra la pantalla para el registro de la Remuneraciones del empleado.
3. El actor procede con el llenado de la información referida a Remuneraciones, entre los cuales se podemos mencionar:
    - Concepto de ingresos (básico, bonificaciones, movilidad, etc.)
    - Importe
    - Tipo de moneda
    - Estado
    - Fecha de vigencia
4. Encargado de Personal presiona el botón “Guardar” para guardar los datos ingresados.

[FA3] El actor decide salir del sistema sin registrar sus datos

1. El actor no ingresa ningún dato y sale de la pantalla donde se encuentra.
2. El caso de uso termina.

(g) Requerimientos Especiales

No aplica.

(h) Puntos de Inclusión

No aplica.

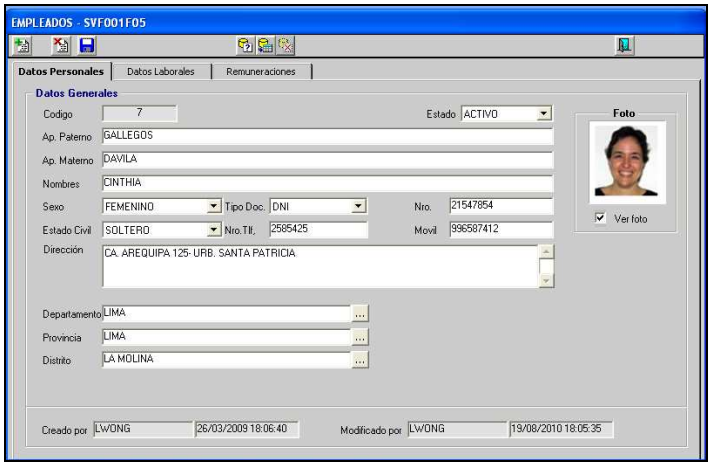
(i) Puntos de Extensión/Herencia

No aplica.

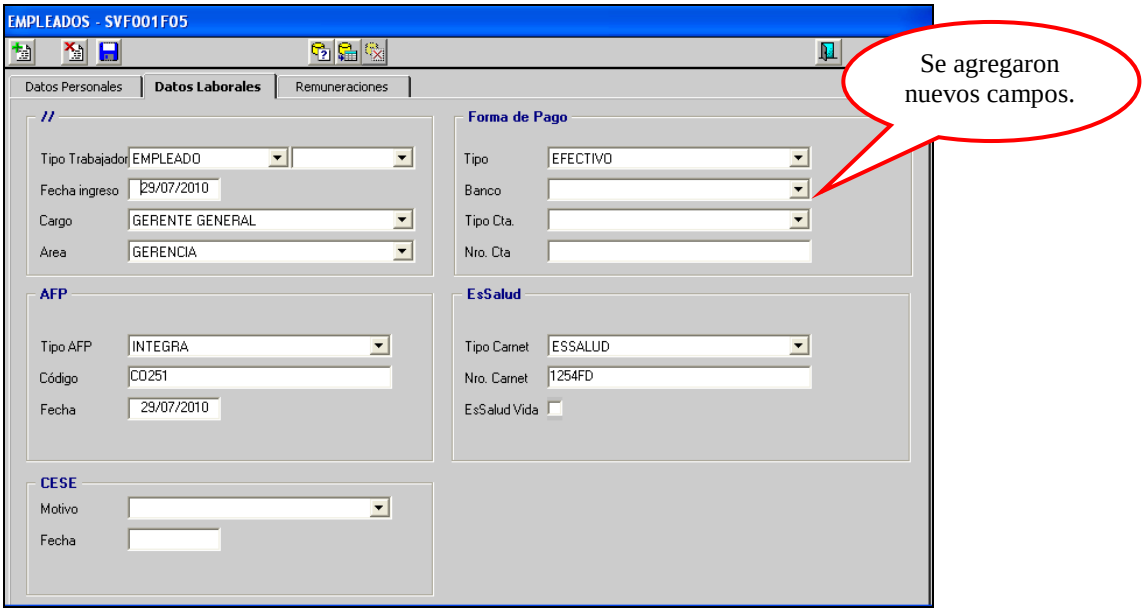
(j) Excepciones

No aplica.

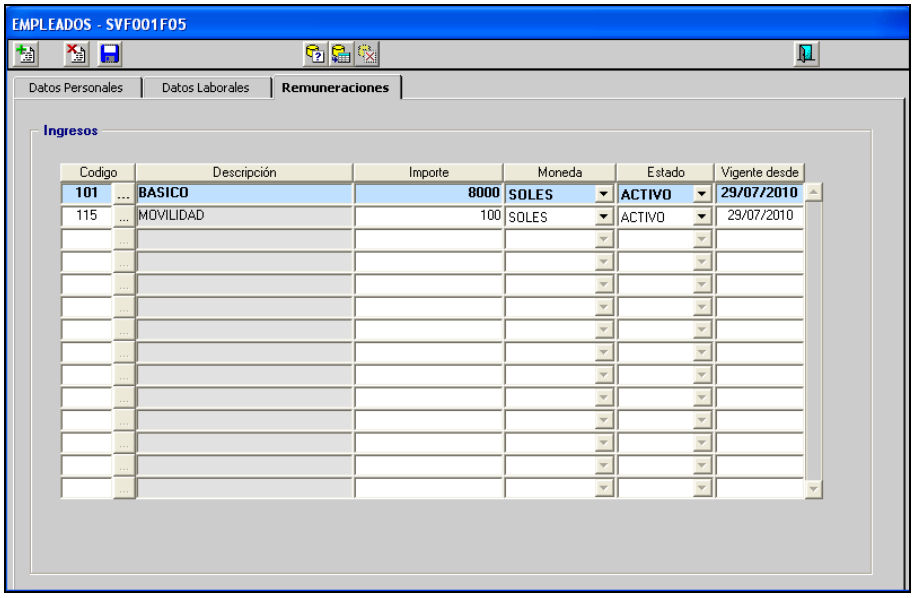
(k) Prototipos del Sistema



Pestaña de registro de Datos Personales.



Pestaña de registro de Datos Laborales.



Pestaña de registro de Datos sobre remuneraciones.